Numéro d'ordre

Année 1973

THESE

Présentée à

L'UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE TROISIEME CYCLE

"Informatique"

par

Jacques LEROUDIER

—o—

une analyse de système

—o—

Thèse soutenue le 10 Avril 1973 devant la commission d'examen

Monsieur N. GASTINEL Président

Messieurs L. BOLLIET
E. GELENBE
M. TERRENOIRE
} **Examinateurs**

G. BOKSENBAUM Invité

Numéro d'ordre

Année 1973

THESE

Présentée à

L'UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE TROISIEME CYCLE

"Informatique"

par

Jacques LEROUDIER

—o—

une analyse de système

—o—

Thèse soutenue le 10 Avril 1973 devant la commission d'examen

Monsieur N. GASTINEL Président

Messieurs L. BOLLIET
E. GELENBE
M. TERRENOIRE
} **Examinateurs**

G. BOKSENBAUM Invité

UNIV.
LEIDEN
BIBL.

UNIVERSITE SCIENTIFIQUE
ET MEDICALE DE GRENOBLE

LISTE DES PROFESSEURS

Président : Monsieur Michel SOUTIF
Vice-Président : Monsieur Gabriel CAU

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Georges	Clinique des maladies infectieuses
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BENOIT Jean	Radioélectricité
	BERNARD Alain	Mathématiques Pures
	BESSON Jean	Electrochimie
	BEZES Henri	Chirurgie générale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BONNIER Etienne	Electrochimie Electrométallurgie
	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques Appliquées
	BRAVARD Yves	Géographie
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	CABANAC Jean	Pathologie chirurgicale
	CABANEL Jean	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et Toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Oto-Rhino-Laryngologie
	CHATEAU Robert	Thérapeutique
	CHENE Marcel	Chimie papetière
	COEUR André	Pharmacie chimique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique

Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	FELICI Noël	Electrostatique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques Pures
	GALVANI Octave	Mathématiques Pures
	GASTINEL Noël	Analyse numérique
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques Pures
	GIRAUD Pierre	Géologie
	KLEIN Joseph	Mathématiques Pures
Mme	KOFLER Lucie	Botanique et Physilogie végétale
MM.	KOSZUL Jean-Louis	Mathématiques Pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LOUP Jean	Géographie
Mle	LUTZ Elisabeth	Mathématiques Pures
MM.	MALGRANGE Bernard	Mathématiques Pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MASSEPORT Jean	Géographie
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et Pétrographie
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAUTHENET René	Electrotechnique
	PAYAN Jean-Jacques	Mathématiques Pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET René	Servomécanismes
	PILLET Emile	Physique industrielle
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REULOS René	Physique industrielle
	RINALDI Renaud	Physique
	ROGET Jean	Clinique de pédiatrie et de puériculture
	SANTON Lucien	Mécanique
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SILBERT Robert	Mécanique des fluides
	SOUTIF Michel	Physique générale

MM.	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLAND François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
M.	VERAIN André	Physique
Mme	VEYRET Germaine	Géographie
MM.	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	BULLEMER Bernhard	Physique
	HANO JUN-ICHI	Mathématiques Pures
	STEPHENS Michaël	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

MM.	BEAUDOING André	Pédiatrie
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BONNETAIN Lucien	Chimie minérale
Mme	BONNIER Jane	Chimie générale
MM.	CARLIER Georges	Biologie végétale
	COHEN Joseph	Electrotechnique
	COUMES André	Radioélectricité
	DEPASSEL Roger	Mécanique des fluides
	DEPORTES Charles	Chimie minérale
	GAUTHIER Yves	Sciences biologiques
	GAVEND Michel	Pharmacologie
	GERMAIN Jean-Pierre	Mécanique
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	HACQUES Gérard	Calcul numérique
	JANIN Bernard	Géographie
Mme	KAHANE Josette	Physique
MM.	MULLER Jean-Michel	Thérapeutique
	PERRIAUX Jean-Jacques	Géologie et Minéralogie
	POULOUJADOFF Michel	Electrotechnique
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	ROBERT André	Chimie papetière
	DE ROUGEMONT Jacques	Neurochirurgie
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIBILLE Robert	Construction mécanique
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Dermatologie
	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Yves	Chimie
	BEGUIN Claude	Chimie organique
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BILLET Jean	Géographie
	BLIMAN Samuel	Electronique (EIE)
	BLOCH Daniel	Electrotechnique
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BOUCHET Yves	Anatomie
	BOUVARD Maurice	Mécanique des fluides
	BRODEAU François	Mathématiques (IUT B)
	BRUGEL Lucien	Energétique
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHIAVERINA Jean	Biologie appliquée (EFP)
	CHIBON Pierre	Biologie animale
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	DURAND Francis	Métallurgie
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROULADE Joseph	Biochimie médicale
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Médecine préventive
	IDELMAN Simon	Physiologie animale
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	JOLY Jean-René	Mathématiques Pures
	JOUBERT Jean-Claude	Physique du solide
	JULLIEN Pierre	Mathématiques Pures
	KAHANE André	Physique générale
	KUHN Gérard	Physique
	LACOUME Jean-Louis	Physique
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LANCIA Roland	Physique atomique
	LE JUNTER Noël	Electronique
	LEROY Philippe	Mathématiques
	LOISEAUX Jean-Marie	Physique nucléaire
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LUU DUC Cuong	Chimie organique
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et Médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)

MM.	MAYNARD Roger	Physique du solide
	MICHOULIER Jean	Physique (IUT A)
	MICOUD Max	Maladies infectieuses
	MOREAU René	Hydraulique (INP)
	NEGRE Robert	Mécanique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PELMONT Jean	Physiologie animale
	PERRET Jean	Neurologie
	PERRIN Louis	Pathologie expérimentale
	PFISTER Jean-Claude	Physique du solide
	PHELIP Xavier	Rhumatologie
Mle	RIERY Yvette	Biologie animale
MM.	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RENAUD Maurice	Chimie
	RICHARD Lucien	Botanique
Mme	RINAUDO Marquerite	Chimie macromoléculaire
MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VAN CUTSEM Bernard	Mathématiques appliquées
	VEILLON Gérard	Mathématiques appliquées (INP)
	VIALON Pierre	Géologie
	VOOG Robert	Médecine interne
	VROUSSOS Constantin	Radiologie
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM.	BOUDOURIS Georges	Radioélectricité
	CHEEKE John	Thermodynamique
	GOLDSCHMIDT Hubert	Mathématiques
	SIDNEY STUARD	Mathématiques Pures
	YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

Mme	BERIEL Hélène	Physilogie
Mme	RENAUDET Jacqueline	Microbiologie

Fait le 30 mai 1972.

Je tiens à remercier :

Monsieur Noël GASTINEL, Professeur à l'Université Scientifique et Medicale de Grenoble, qui a bien voulu me faire l'honneur de présider le jury de cette thèse et qui s'est beaucoup intéressé à mes résultats,

Monsieur Louis BOLLIET, Professeur à l'Université Scientifique et Médicale de Grenoble, qui m'a toujours encouragé dans mes recherches,

Messieurs Erol GELENBE, Ingénieur de Recherche à l'IRIA, et Michel TERRENOIRE, Maître de Conférence à l'Université de Lyon I, qui se sont intéressés à mon travail et ont bien voulu accepter de le juger,

Monsieur Claude BOKSENBAUM, Attaché de Recherche au CNRS, qui m'a constamment guidé et conseillé et à qui je dois d'avoir pu faire cette thèse,

L'équipe du système CP/CMS qui, malgrè les tracas que je lui ai causés, m'a beaucoup aidé,

Tous mes camarades, amies et amis qui m'ont apporté leur sympathie, leurs encouragements et leur aide : Dominique BORRIONE et François SCHWAAB qui ont pris une part active à ce travail, Jacques BRIAT qui m'a permis de décanter mes idées au cours de longues croisières, Anne-Marie CHAVANNE, Pierre LAFORGUE, Luc FINET, Dominique TONNELLIER et bien d'autres encore,

Le service de dactylographie qui a assuré la frappe de mon manuscrit avec compétence,

Le service de reproduction qui a réalisé un travail remarquable.

Ce travail a été financé par les contrats D.R.M.E.

(Direction des Recherches et Moyens d'Essais)

n° 70-34-179.00.480.75.01

et

n° 72.34.601.00.480.75.01

TABLE DES MATIERES

Pages

INTRODUCTION 1

CHAPITRE 1 : LES MESURES ET LES OUTILS DE PRELEVEMENT

1.1. GENERALITES 5
Le modèle
Les techniques

1.2. CE QUI EST POSSIBLE DANS UN CONTEXTE MONOPROCESSEUR 11
1.2.1. Au niveau du hardware 11
1.2.2. Au niveau du software 11

1.3. CE QUI EST POSSIBLE DANS UN CONTEXTE MULTIPROCESSEUR 12
1.3.1. Au niveau du hardware 12
1.3.2. Au niveau du software 15

1.4. CONCLUSIONS 16
Adaptation à l'utilisation
Notion de systèmes auto-adaptatifs

CHAPITRE 2 : MESURES ET OUTIL REALISE DANS CP/CMS 17

2.1. LE SYSTEME 17
2 niveaux pour l'outil

2.2. L'OUTIL AU NIVEAU DE CP 19
2.2.1. Présentation de CP 19
Machine virtuelle 19
Le "Spooling" 21
Démarrage à chaud ou à froid 24
2.2.2. Les moyens de communications entre machines virtuelles 25
2.2.3. La communication réalisée 26

Pages

2.3. L'OUTIL AU NIVEAU DE CMS 28
2.3.1. Présentation de CMS 28
Configuration mémoire 30
Les commandes CMS 31
Les macro-commandes 32
2.3.2. Prises de mesures dans CMS 36

2.4. FONCTIONNEMENT GENERAL 39
2.4.1. La machine CMSMESUR 39
2.4.2. Fonctionnement du point de vue de l'opérateur 40

CHAPITRE 3 : LES PREMIERS RESULTATS

3.1. CONFIGURATION ET ENVIRONNEMENT DE L'IMAG 42
La configuration
Caractéristiques des échantillons traités 43

3.2. FREQUENCES - MOYENNES - HISTOGRAMMES 45
3.2.1. Généralités 45
Valeurs centrales 45
Mesures de dispersion 46
3.2.2. Résultats pour CP/CMS 48
Groupes logiques
Fréquences
Histogrammes 52
Applications de ces résultats 61
3.2.3. Résultats pour OS/360 - MVT 61
Fréquences
Valeurs moyennes

3.3. COMPARAISONS DE CP/CMS, OS/360-MVT et CTSS 64
3.3.1. CP/CMS et OS/360-MVT 64
3.3.2. CP/CMS et CTSS 65

Pages

CHAPITRE 4 : CORRELATIONS ET REGRESSIONS

4.1. GENERALITES 67
Notion d'ensemble instantané des ressources 67

4.2. CORRELATIONS 68
4.2.1. Rappels 68
4.2.2. Interprétation géométrique 70
4.2.3. Applications et résultats 72
4.2.4. Conclusions 77

4.3. REGRESSIONS 78
4.3.1. Le modèle 78
4.3.2. Les buts et les critères associés 82
But explicatif 82
But prédictif 82
Critères associés 83
4.3.3. Les résultats et interprétations 83
4.3.4. D'autres méthodes d'analyse 87
Création de variables formelles 87
Création de "types" formels 87
Application de la typologie 87

CONCLUSION 90

BIBLIOGRAPHIE 95

ANNEXES 101

INTRODUCTION

L'historique de toute conception humaine comporte deux phases :

- réalisation de la fonction désirée
- amélioration de la réalisation suivant des critères spécifiques.

Les systèmes informatiques n'ont pas échappé à cette règle.

Rapidement, après avoir fait des systèmes qui "marchent", on a voulu faire faire des systèmes qui "marchent" le mieux possible suivant des critères tels que la place prise en mémoire centrale et le temps d'exécution des programmes.

Les améliorations que l'on a pu obtenir ont tout d'abord été inconditionnelles. Il est certain que des programmes équivalents plus courts, des composants électroniques et des accès mémoire plus rapides sont, dans tous les cas, des améliorations. Mais après ce stade, dont les limites sont fréquemment la technologie et le coût financier, il faut en arriver à des améliorations conditionnelles. C'est-à-dire des améliorations qui sont liées aux comportements internes du système ou à l'utilisation qui en est faite. Ainsi, l'introduction des mémoires à "cache" (on parle aussi d'anté-mémoire) ou des mémoires à adressage paginé n'est une amélioration que si on admet qu'un programme, lors d'un court intervalle de temps, n'utilise qu'une partie de l'espace adressable ("localité"). Une telle amélioration est conditionnée par le comportement des programmes tant système qu'utilisateur, car il est possible d'écrire des programmes tels que le "cache"

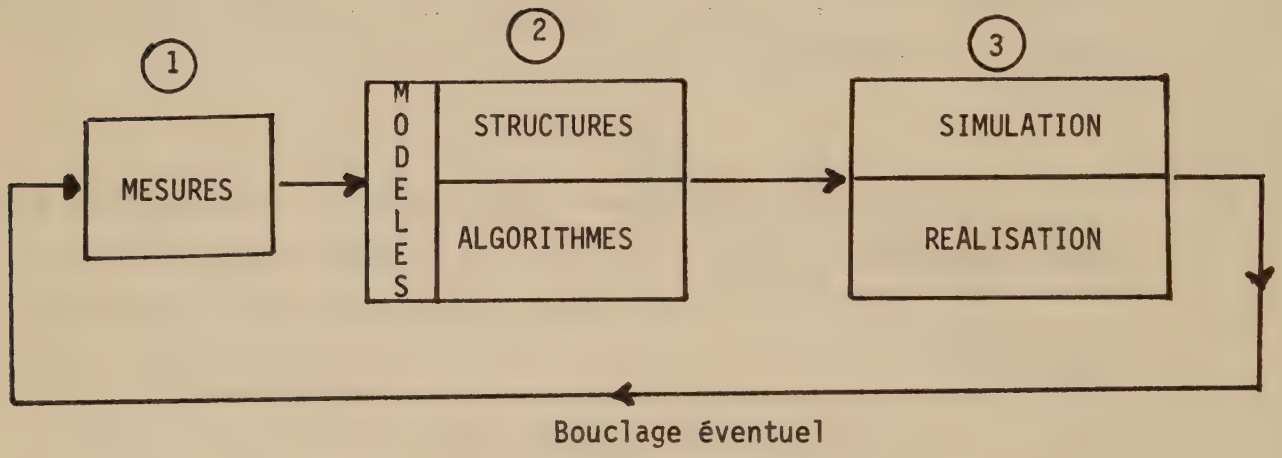
ou la pagination n'apportent aucune amélioration.

Il est donc très important de savoir si une modification ou une nouveauté est une amélioration. Ce problème ne peut être résolu que par la prise de mesures et leur analyse.

Les mesures permettent :

- de juger des améliorations apportées
- d'éviter de fausses améliorations
- de déterminer celles qui sont à faire.

Aussi, lors de l'étude ou de la conception d'un système, la démarche généralement adoptée comprend 3 étapes qui inter-agissent sur elles-mêmes et les unes sur les autres.



(1) Mesures : au premier tour dans le schéma, elles proviennent d'idées à priori et, d'estimations liées au matériel employé.
Aux deux autres tours, elles proviennent
- soit de simulations
- soit de versions précédentes du système

(2) Modèles : Ils sont constitués des structures du système et des algorithmes qui animent et gèrent ces structures.
Ces modèles peuvent être multiples et plus ou moins détaillés suivant qu'ils sont destinés à être simulés ou réalisés.

(3) Simulation - réalisation : Avant de passer à la réalisation ou simultanément avec elle, on simule le système envisagé.
Eventuellement lors d'études théoriques, par manque ou économie de moyens, le système n'est jamais concrètement réalisé, mais toujours simulé.

L'étude qui a été faite porte sur l'étape des mesures et débouche sur celle des modèles puisqu'elle remet en cause des structures et des algorithmes.

Les améliorations recherchées sont conditionnelles, ces améliorations étant conditionnées non plus par des comportements internes du système, mais par des comportements de l'utilisateur à un niveau symbolique, c'est-à-dire au niveau de l'emploi de "processeurs" (compilateurs, éditeurs, chargeurs, etc...).

Les problèmes abordés par ce travail sont liés aux mesures :

- les mesures et les outils pour les recueillir
- la réduction de données
- l'analyse et l'interprétation de données
- la comparaison de données

et débouchent sur des conceptions nouvelles de systèmes : les systèmes auto-adaptatifs qui possèdent des outils intégrés de mesures inter-agissant

sur le système lui-même.

Cette auto-adaptation sera peut-être un premier pas vers l'élaboration de systèmes à apprentissage, qui, non contents de se modifier en fonction de l'utilisation qui en est faite, analyseront les situations passées pour pouvoir mieux les appréhender dans le futur.

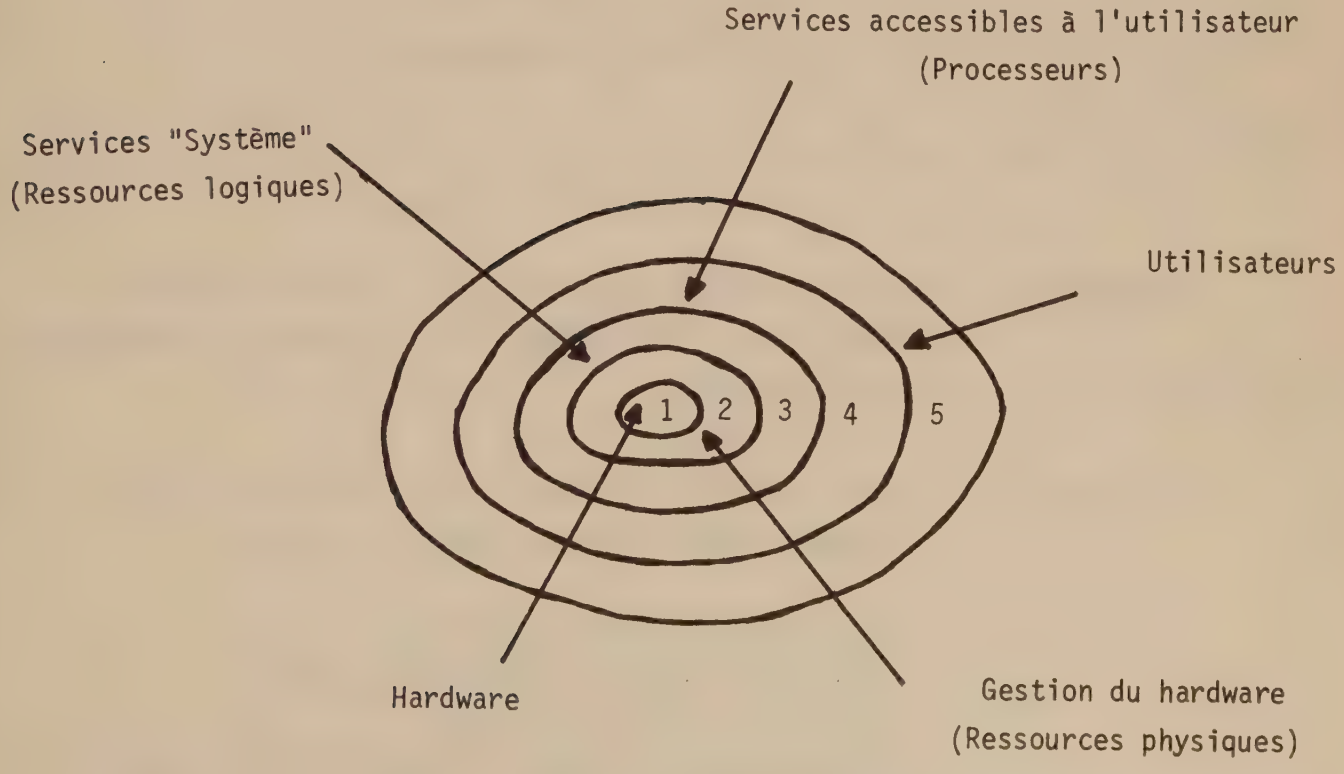
C H A P I T R E 1

<><******><>

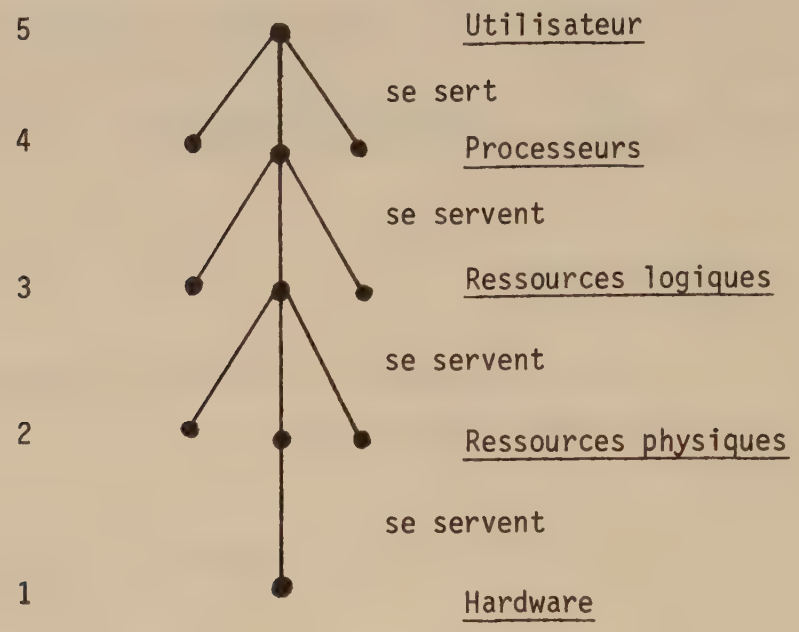
1. LES MESURES ET LES OUTILS DE PRELEVEMENT

1.1. Généralités

Nous nous proposons d'étudier un système selon le modèle d'analyse en couches logiques suivant :



Cette notation représente la structure logique que nous mettons sur un système :



La couche 1 est constituée du hardware de la machine dont une partie des fonctions sont éventuellement réalisées par microprogrammes.

La couche 2 gère le hardware de la machine. C'est à ce niveau, selon ce dont on dispose, que se trouvent :

- le lancement des E/S physiques
- la constitution des programmes canaux
- la gestion des interruptions
- la gestion de la mémoire (mémoire libre, clés de protection, adressabilité, registres de base, pagination, etc...).

La couche 3 comprend les ressources logiques du système. Ainsi, à ce niveau, les E/S sont logiques : les interruptions et les programmes canaux (s'ils existent sur la machine) sont inconnus.

La couche 4 est constituée des processeurs utilisables :

- éditeurs
- compilateurs
- chargeurs
- modules permettant à un utilisateur d'exécuter ses programmes
- etc...

Les problèmes "délicats" dans un système (pagination, E/S, interblocages, etc...) se trouvent dans la couche 2. Ils ont été étudiés par des outils de mesures placés successivement dans les couches 2 et 3, ce qui a permis des solutions locales à ces niveaux. Cette démarche de remontée dans les couches est particulièrement claire pour les algorithmes de pagination [36] qui furent :

- FIFO (First-In First-Out simple file d'attente)
  couche 2
- LRU (Least Recently Used - page n'ayant pas été référencée depuis le plus longtemps)
  couche 2
- "Working set" [36,10] couches 3 et 4.

Nous nous proposons d'effectuer la démarche inverse, c'est-à-dire une descente des couches externes vers les couches internes, pour essayer de prévoir, avant d'arriver dans une couche, les problèmes qui vont s'y poser. Il faut, en effet, remarquer que plus on est haut dans la structure, plus les problèmes se posent à un niveau logique où ils n'ont pas le caractère impératif qu'ils peuvent avoir à un niveau physique.

Une telle étude exige des outils pour prélever des mesures dans les diverses couches et des traitements de données associés.

Ces outils et ces traitements ne sont pas deux problèmes indépendants. Ainsi la réduction des données n'est pas exclusivement un problème mathématique. En effet, lors de l'acquisition des données, il y a une réduction implicite qui dépend :

- des variables enregistrées
- des techniques de prélèvement employées

Les techniques de prélèvement sont de 3 sortes :

a - Echantillonnage

Les données désirées sont prélevées sur l'apparition d'un évènement extérieur. Cet évènement extérieur peut être indépendant du fonctionnement du système :

- signaux d'horloge
- décision humaine (appuyer sur un bouton)
- passage par certaines valeurs de compteurs d'évènements externes,

ou, au contraire, dépendre du fonctionnement du système

- passage par certaines valeurs de compteurs d'évènements internes.

La réduction implicite effectuée par de tels outils dépend :

- du pas de l'échantillonnage (signaux d'horloge plus ou moins serrés, etc...)
- du pouvoir de résolution de l'outil (ainsi un outil hardware dont le temps de réaction est de l'ordre de la microseconde ne pourra pas prendre en compte des manipulations de données de registre à registre).

b - Trace

Cette technique consiste à enregistrer un ou plusieurs évènements choisis (avec éventuellement des paramètres qui leur sont associés) chaque fois qu'ils se produisent.

Si l'évènement choisi est l'exécution d'une instruction machin toutes les instructions exécutées seront enregistrées avec éventuellement l'adresse de l'instruction et les adresses des paramètres.

Si l'évènement choisi est l'exécution des commandes système, toutes les commandes seront enregistrées avec éventuellement les caractéristiques des fichiers manipulés.

La technique de trace comparée à la technique d'échantillonnage consiste à enregistrer les variables choisies en continu, au lieu de les enregistrer de temps à autre.

c - Instrumentation directe

Il ne s'agit pas là d'une technique fondamentalement différente des deux précédentes, si ce n'est que les mesures sont prélevées par des outils spéciaux (tant hardware que software), créés spécifiquement pour les variables concernées.

Ainsi au niveau hardware, on étudiera les instructions exécutées en se branchant directement sur les circuits de décodage des instructions tandis qu'au niveau software, on étudiera l'utilisation des routines système en y plaçant des "trappes" pour détecter leur exécution.

Une fois les mesures acquises, les traitements mathématiques des données doivent permettre d'exploiter au mieux les possibilités de l'outil de prélèvement en fonction des buts recherchés.

Ces traitements suivent généralement 2 grandes directions :

a - Réduction des données

Pour permettre leur interprétation les variables sont réduites à des grandeurs caractéristiques telles que moyennes, médianes ou écarts-type et approchées par des lois théoriques. Le nombre des variables peut même être réduit aux variables essentielles à l'explication du phénomène étudié.

b - Reconstitution de variables

Si l'outil de prélèvement n'a pas fourni directement les variables désirées, elles sont reconstituées par des méthodes mathématiques du type Monte Carlo ou par des simulations [17] .

Par exemple, si on ne peut pas tracer toutes les instructions exécutées par une machine, mais qu'on a la possibilité de connaître toutes les t secondes l'instruction en cours d'exécution et les adresses référencées, sous certaines hypothèses de continuité, on peut déterminer la fréquence d'utilisation des zones mémoire, bien que ces variables n'aient pas été directement mesurées.

L'outil de mesure disponible est essentiel, car il détermine non seulement les variables qu'il est possible de mesurer, mais aussi une partie des traitements mathématiques à affectuer.

La classification habituelle des outils de mesure distingue :

- les outils hardware où il n'y a pas ou peu de modifications à apporter au système mesuré et où il n'y a pas ou peu de distorsions dûes à l'outil.
- les outils software qui imposent des modifications du système mesuré et où les distorsions peuvent ne pas être négligeables.

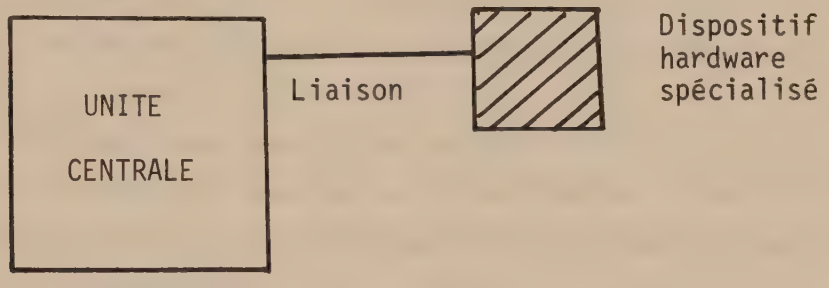
Cette classification est basée sur la technologie et la distorsion éventuelle introduite, mais ne reflète pas les possibilités

et la puissance de l'outil. Nous préférons faire la distinction de ce qu'il est possible de faire dans un contexte monoprocesseur ou multiprocesseur.

1.2. Ce qui est possible dans un contexte monoprocesseur

1.2.1. Au niveau hardware

Une méthode classique consiste à brancher directement sur les circuits de l'unité centrale un dispositif hardware spécialisé. Celui-ci prélève directement des informations sur la machine, sans en gêner le fonctionnement.



Ces informations peuvent être combinées par des fonctions logiques ("ET", "OU", etc...) avant d'être transmises à des compteurs :

- soit d'évènements
- soit de signaux d'horloge

Le contenu de ces compteurs peut être soit visualisé, soit envoyé sur un support externe pour un traitement différé.

C'est ce qui est réalisé chez IBM avec le Basic Counter Unit (BCU) [13]. Il est, par exemple, possible de connaître le temps d'attente de l'unité centrale avec un canal occupé ("ET" entre les évènements: Unité Centrale en attente, canal occupé, signaux d'horloge).

REMARQUE

> De tels outils n'exigent aucune modification du système, mais sont inaccessibles de ce dernier.

1.2.2. Au niveau software

Les outils software se divisent en 2 parties bien distinctes.

a - L'outil lui-même

qui prélève les mesures et effectue éventuellement un pré-traitement. Suivant l'implémentation adoptée il peut être dans le système lui-même (importantes modifications) ou à l'extérieur du système, mais avec un mécanisme pour lui donner le contrôle (outil placé en "verrue" du système).

b - Le mécanisme qui donne le contrôle à l'outil

peut être inexistant si l'outil est totalement intégré au système. Sinon le passage de contrôle est fréquemment assuré par des instructions du type "appel superviseur" (Supervisor Call des 360 IBM, CALn du 10070 CII ou même les instructions masquables Monitor Call des 370 IBM). A défaut de telles instructions, on provoque parfois des interruptions programme (par exemple : code d'instruction inexistante ou privilégiée) auxquelles on donne une signification particulière.

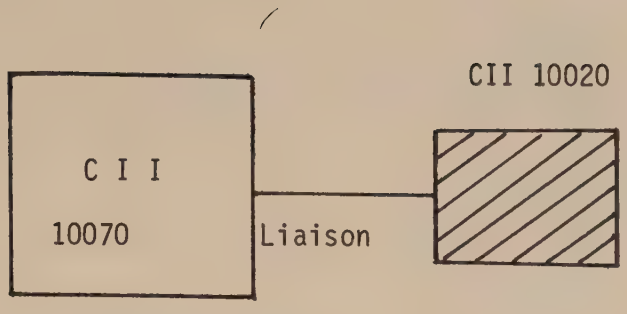
Il est clair que ces outils exigent toujours un minimum de modifications du système mesuré. Par ailleurs ils peuvent introduire des distorsions dans le temps non négligeables.

La plupart des outils expérimentaux sont de ce type [17,30], et il en existe même qui sont distribués avec une version standard de système (par exemple SMF dans OS/360 d'IBM [22]).

1.3. Ce qui est possible dans un contexte multiprocesseur

1.3.1. Au niveau hardware

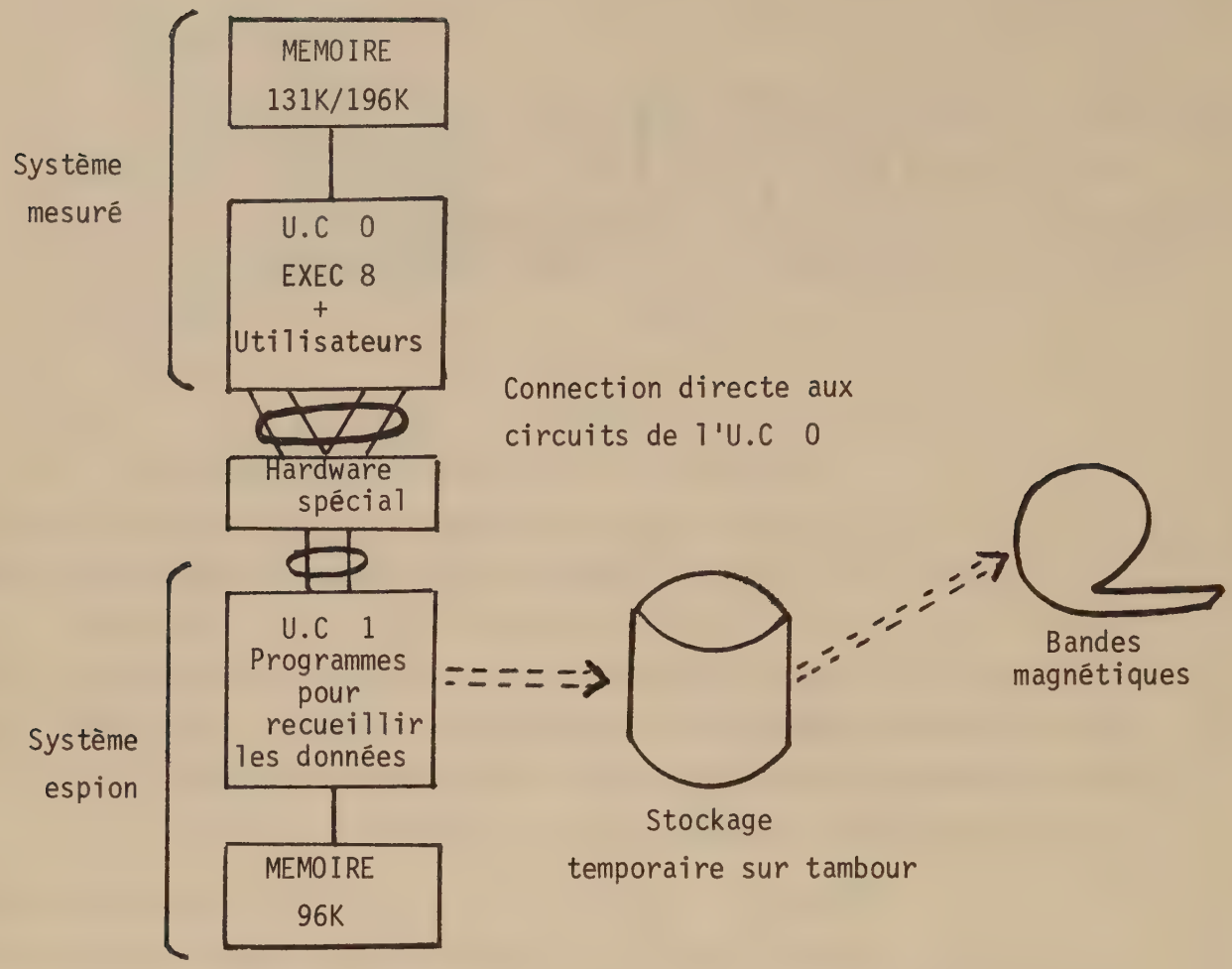
Il s'agit souvent d'une extension du cas monoprocesseur: on perfectionne tellement l'unité hardware connectée sur la machine qu'on en arrive à utiliser un petit ordinateur.



L'avantage d'une telle méthode est que le petit calculateur peut effectuer un pré-traitement relativement important des données.

Il faut remarquer que si on veut obtenir des mesures analogues à celles fournies par un BCU, on doit disposer d'un hardware supplémentaire permettant au petit calculateur d'avoir accès aux circuits de la machine mesurée.

BORDSEN a réalisé, dans cet esprit, une expérience intéressante sur UNIVAC 1108 [6]. Il dispose de deux UNIVAC 1108, dont l'un espionne l'autre, et d'un hardware spécial permettant à l'unité centrale espion d'accéder aux circuits de l'unité centrale espionnée qui tourne sous le système standard EXEC 8 :



Les informations recueillies sont de 3 types :

- trace des programmes exécutés
- trace des instructions exécutées
- trace des adresses mémoire référencées

Ce qui permet d'étudier l'utilisation de la mémoire, les séquences d'instructions et la localité des programmes.

Il faut noter qu'un tambour est saturé au bout de 5 à 20 secondes de prise de mesures!

1.3.2. Au niveau software

L'avantage d'être dans un contexte multiprocesseur est de pouvoir relier facilement les niveaux hardware et software, donc les diverses couches dans notre modèle de système.

Dans le cas de BORDSEN, comme le système espion connait le système espionné, la connaissance des références mémoire entraine celle des modules utilisés.

Une structure originale de système multiprocesseur est le Control Data 6600 qui possède un calculateur central et 10 calculateurs périphériques. Chaque calculateur dispose d'une mémoire et les calculateurs périphériques ont leur propre code d'instructions. Chaque calculateur périphérique peut lire et écrire à tout moment dans la mémoire du calculateur central et interrompre ce dernier.

De nombreuses mesures ont été effectuées sur cette machine avec des systèmes d'exploitation différents [33,34]. Toutes les expériences faites avaient pour principe de spécialiser un des calculateurs périphériques dans la prise et l'analyse de mesures. SHERMAN [33] utilise les résultats obtenus à des algorithmes de scheduling prédictifs.

Il faut noter qu'on aboutit à une notion d'outil de mesure intégré : l'outil fait partie du système, il ne constitue pas quelque chose d'indépendant ou de rajouté.

1.4. Conclusions

Cet aperçu des différentes techniques et méthodes de mesures des systèmes montre que les outils de prélèvement et les traitements, s'ils sont très liés, sont toujours séparés :

- d'une part on recueille des informations
- d'autre part on les traite en différé.

Dans tous les cas décrits, sauf celui d'une seule unité centrale avec BCU, il était techniquement possible que les informations prélevées interagissent sur le système. Nous reviendrons plus loin sur cette notion de systèmes auto-adaptatifs pour la justifier et indiquer les outils hardware et software qu'elle suppose.

Il faut remarquer que dans toutes les études qui ont été faites, sauf celle de SCHERR [32], on ne s'est jamais inquiété du comportement de l'utilisateur pour voir l'influence qu'il avait sur le système.

C'est cette préoccupation qui a guidé notre travail. Nous décrivons dans les chapitres qui suivent ce qui a été fait et pour commencer l'outil de prise de mesures que nous avons dû implémenter.

CHAPITRE 2

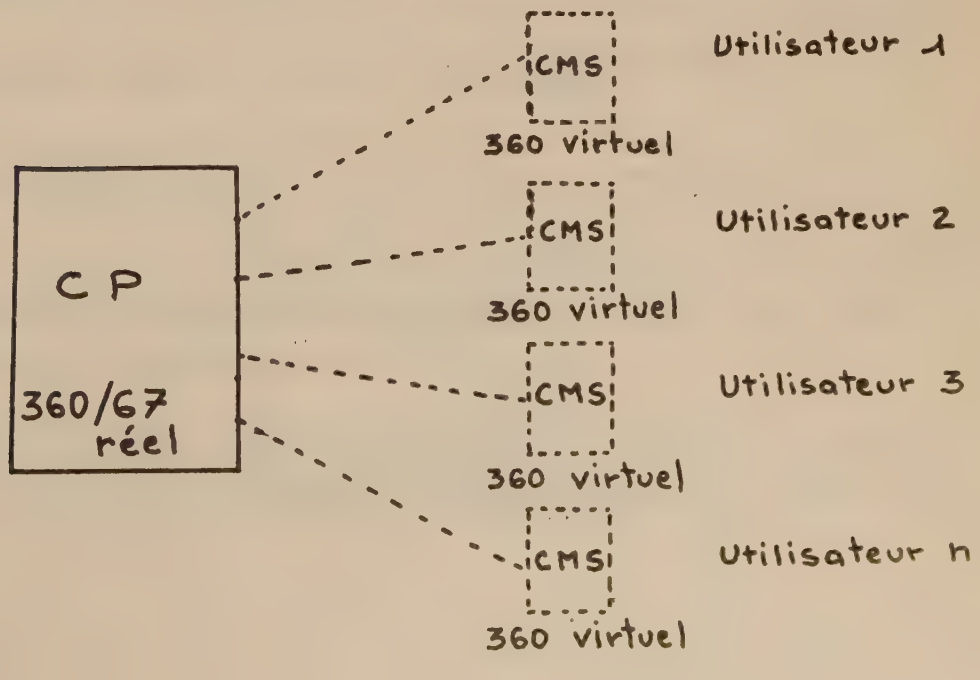
<><******><>

2. MESURES ET OUTIL REALISE DANS CP/CMS

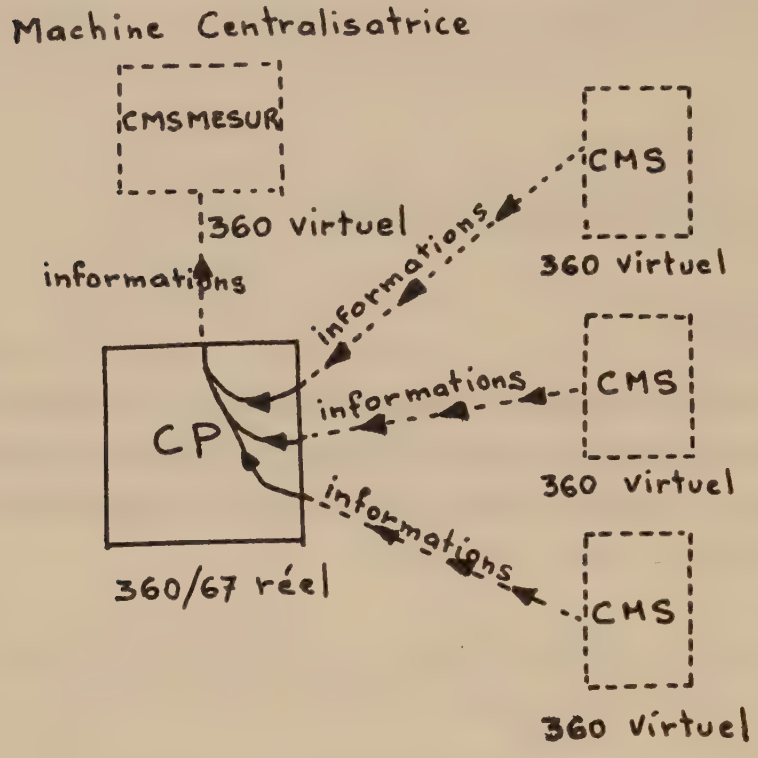
2.1. Le système [2]

Le système choisi pour cette étude est le système conversationnel en temps partagé CP/CMS en service à l'Institut d'Informatique et de Mathématiques Appliquées de Grenoble sur un ordinateur IBM 360/67 de 1024 K octets de mémoire centrale.

Dans le système CP/CMS, la partie "temps partagé" est assurée par CP qui génère pour chaque utilisateur un 360 virtuel (simulation d'un 360 IBM) qu'il active périodiquement et la partie "conversationnelle" par CMS qui est "monté" dans chaque 360 virtuel généré par CP. Il faut remarquer que CP et CMS peuvent "tourner" indépendamment l'un de l'autre. En effet, on pourrait "monter" un autre système que CMS dans un 360 virtuel (OS par exemple) ou encore "monter" CMS sur un 360 réel.



L'idée de base de l'outil est d'avoir dans chaque machine virtuelle un système CMS modifié pour émettre de l'information vers une machine centralisatrice des données (machine CMSMESUR).



Cet outil est donc à 2 niveaux :

a- *Niveau CP* : communication entre machines virtuelles pour recueillir les mesures faites

b- *Niveau CMS*: recherche et prise des mesures voulues

Pour devenir opérationnel, il devra satisfaire aux contraintes suivantes:

- fiabilité
- transparence vis à vis des utilisateurs (en n'exigeant rien de leur part et en ne modifiant pas sensiblement leur temps de réponse)
- simplicité d'emploi vis à vis des opérateurs du système (pour lancer ou arrêter la prise des mesures).

2.2. L'Outil au niveau de CP

2.2.1. Présentation de CP [19]

Le système CP-67 (Control program - 67) est un système "temps partagé" destiné à "tourner" sur un ordinateur IBM 360/67 (le modèle 67 de la série IBM 360 se distingue par un dispositif spécial d'adressage de la mémoire, dit de "traduction dynamique des adresses") [20]. Le partage du temps est effectué en simulant pour chaque utilisateur un ordinateur IBM 360 (génération de machines virtuelles) et en activant périodiquement cet "ordinateur". L'utilisateur "charge" alors dans sa machine virtuelle le système qui lui semble le plus approprié à son type de problème (CMS, dans un contexte conversationnel et intéractif).

Machine virtuelle

La "technologie virtuelle" fournie par CP pour créer une machine virtuelle est la suivante :

- une mémoire continue adressable normalement où les emplacements ayant une signification technologique (horloge, double-mots pour échange de mots d'état lors des interruptions) sont simulés.Ceci est rendu possible grâce au dispositif de "traduction dynamique des adresses" (On parle souvent de pagination) et aux deux modes de fonctionnement de la machine (mode "maître" et mode "esclave").

- une configuration d'unités :

  . semi-virtuelles, comme les disques. Une unité de disque pour une machine virtuelle sera en réalité une portion de disque réel (quelques cylindres : 5, 20 ou 30 cylindres sur les 203 d'un disque réel);

  . virtuelles comme les les lecteur, perforateur, imprimante, etc... où la simulation de telles unités est assurée par "spooling" ;

. réelles lorsqu'il s'agit d'unités non partageables ou non simulables "on line", comme les dérouleurs de bande ou un "display". De telles unités ne peuvent pas faire partie de la configuration standard d'une machine virtuelle, mais elles peuvent être"attachées" à telle ou telle machine (utilisateur) par l'opérateur du système (c'est donc lui qui assure le "partage" de ces unités).

Voyons maintenant concrètement, comment CP réalise cette "technologie virtuelle".

A chaque machine virtuelle est associée une table (UTABLE) qui décrit complètement sa configuration et son état :

- à chaque désactivation de la machine (pour épuisement de sa tranche de temps ou pour d'autres raisons), les registres généraux et flottants et le mot d'état sont préservés dans cette table pour permettre son relancement;

- un pointeur (qui servira à charger le registre de contrôle 0 du 360/67) pointe vers la table des "segments" constituée de pointeurs vers les tables de "pages" de la machine. C'est cet ensemble de tables et de pointeurs qui, grâce au mécanisme de pagination du 360/67, crèe et décrit la mémoire virtuelle de la machine.

- un ensemble de pointeurs vers des chaînes de tables qui décrivent les unités virtuelles ou réelles attachées à la machine et l'état de ces unités (E/S en cours, "fin sur canal", "fin sur unité", etc...).

Le fonctionnement général de CP est alors le suivant :

- seul CP fonctionne en mode "maître",

- Les machines virtuelles fonctionnent en mode "esclave" (instructions privilégiées interdites) même lorsqu'elles croient être en mode "maître",

- Les emplacements réservés en mémoire basse donnent le contrôle à CP lors des interruptions.

Comme on simule un360 sur un 360, CP laisse exécuter librement les instructions non privilégiées par les machines virtuelles et il reprendra le contrôle sur interruption programme lorsque l'une d'entre elles essaiera d'exécuter une instruction privilégiée. Cette instruction privilégiée sera alors :

- soit simulée si la machine virtuelle croit être en mode "maître" (pseudo mode "maître"),

- soit ignorée et provoquera la simulation d'une interruption programme, dans le cas contraire.

A chaque interruption CP entretient l'état des machines concernées (UTABLE et tables d'unités) ainsi que l'état de leur mémoire basse virtuelle.

Le spooling

La technique du SPOOLING (Simultaneous Peripherical Operation On Line) a été utilisée d'une manière générale dès qu'on a voulu optimiser les Entrées/Sorties d'unités lentes (lecteur, perforateur de cartes, imprimante). Elle consiste à mettre en tampon d'unités lentes des unités plus rapides telles que des disques :

- les cartes sont lues en avance et mises sur disque et c'est sur disque qu'on ira les chercher lorsqu'on en aura besoin;

- les sorties, soit sur perforateur, soit sur imprimante, sont différées et mises sur disque. Des routines associées au perforateur et à l'imprimante viennent, à la vitesse de ces unités lentes, prendre sur disque les données à sortir.

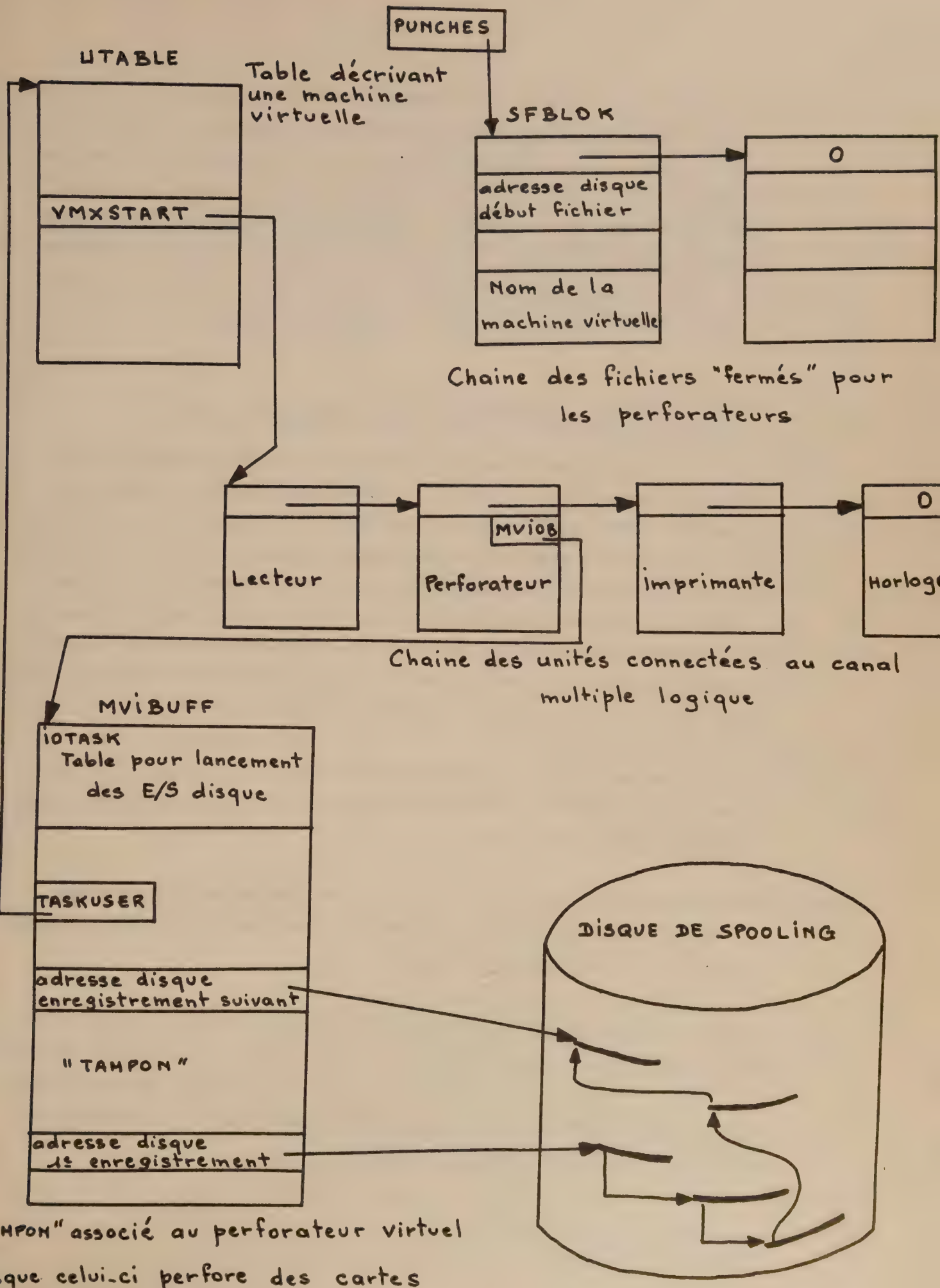
Tout le mécanisme est essentiellement asynchrone et dirigé par les interruptions d'E/S. Cette technique se retrouve dans tous les systèmes multiprogrammés (on parle de tâches parasites ou de symbionts d'entrée-sortie) où elle est indispensable.

Dans CP, cette technique est un peu plus sophistiquée car on vise à donner, par son intermédiaire, des imprimantes, des lecteurs et des perforateurs à chaque machine virtuelle (ces unités étant d'un emploi fréquent, ne serait-ce que pour les entrées et les sorties de compilateurs).

Ainsi, lorsqu'une machine virtuelle croira écrire sur son imprimante (instruction SIO sur l'imprimante), en réalité la ligne à sortir sera mise dans un "buffer" (mémoire tampon) lié à la table décrivant l'imprimante virtuelle de la machine et lorsque ce "buffer" sera plein (au minimum 6 lignes d'imprimante ou 10 cartes, éventuellement plus, car il y a compression des caractères qui se répètent -en particulier les "blancs") il sera écrit sur disque (la machine virtuelle est désactivée durant l'E/S disque). Les "buffers" d'un même fichier sont chaînés entre eux sur disque et lorsque la machine virtuelle aura fini de se servir de son imprimante ("fermeture" de l'imprimante), on chaînera alors un bloc (FBLOK) décrivant ce fichier sur un pointeur (PRINTERS) représentant toutes les imprimantes virtuelles.

Par la suite, les routines associées aux imprimantes réelles de l'installation viendront prendre dans cette chaîne les fichiers à sortir sur imprimante.

Le fonctionnement d'un perforateur virtuel est analogue à celui d'une imprimante, sauf que les blocs décrivant les fichiers "fermés" sont chaînés à partir du pointeur PUNCHES.


PUNCHES
UTABLE
Table décrivant une machine virtuelle
SFBLOK
VMXSTART
adresse disque début fichier
Nom de la machine virtuelle
0
Chaine des fichiers "fermés" pour les perforateurs
Lecteur
Perforateur
MVIOB
imprimante
0
Horlog
Chaine des unités connectées au canal multiple logique
MVIBUFF
IOTASK
Table pour lancement des E/S disque
TASKUSER
adresse disque enregistrement suivant
" TAMPON "
adresse disque 1e enregistrement
DISQUE DE SPOOLING
HPON" associé au perforateur virtuel
que celui-ci perfore des cartes

Pour un lecteur virtuel, le mécanisme est un peu différent. Les cartes sont d'abord lues par le lecteur physique de l'installation (la lecture est provoquée par l'interruption qui survient lorsqu'on met le mecteur "prêt") et mises sur disque en "buffers" chaînés, lorsque le fichier a été entièrement lu, un SFBLOK le décrivant est chaîné sur le pointeur READERS. Quand la machine virtuelle voudra lire ses cartes, on parcourra la chaîne à partir de READERS pour voir s'il y a un fichier pour elle et,dans l'affirmative, on lira ce fichier sur disque.

Arrêt du système - Démarrage à chaud - Démarrage à froid

L'opérateur du système dispose de la commande SHUTDOWN pour arrêter le système CP. Comme la technique de spooling désynchronise les E/S réelles et virtuelles, il se peut très bien (et c'est pratiquement toujours le cas), que l'imprimante réelle n'ait pas fini d'imprimer les sorties des imprimantes virtuelles (le cas est analogue pour les autres unités : perforateurs, etc...). Aussi, à l'arrêt du système, non seulement préserve-t-on dans une zone disque réservée, les informations comptables pour facturation, mais aussi toutes les chaînes qui partent des pointeurs : READERS, PUNCHES, PRINTERS.

Ainsi, lors du démarrage du système, on pourra retrouver où on en était dans le fonctionnement des unités virtuelles. On dira qu'on a fait un démarrage à chaud.

On a aussi la possibilité de redémarrer le système en réinitialisant le "spooling". Tous les fichiers des unités virtuelles sont alors perdus. On parle alors de démarrage à froid.

En cas de cassure, soit parce que CP a détecté une erreur fatale, soit sur interruption programme en mode "maître" (seul CP est en mode "maître" il s'agit donc d'une erreur dans CP lui-même et non chez un utilisateur) il y a un SHUTDOWN forcé; avec un redémarrage à chaud, on peut donc récupérer les fichiers "fermés" du spooling.

2.2.2. Les moyens de communication entre machines virtuelles

Le seul véritable moyen standard de communication entre machines virtuelles est réalisé par l'intermédiaire du "spooling". Grâce à la commande XFER, on peut indiquer au système CP que les sorties d'un perforateur virtuel au lieu d'être dirigées vers un perforateur réel doivent l'être vers l'entrée d'un lecteur virtuel. Bien sûr la commande XFER a en paramètres le perforateur et le lecteur concernés.

Cette liaison entre perforateur et lecteur virtuels est relativement simple à réaliser techniquement, étant donné les mécanismes de fonctionnement du spooling de CP. Lorsque le perforateur virtuel a fini de perforer les cartes qu'on veut transmettre (fermeture du fichier), le bloc décrivant le fichier fermé (SFBLOK), au lieu d'être chaîné sur le pointeur PUNCHES, l'est sur le pointeur READERS. Ainsi lorsque le lecteur à qui l'on transmet voudra lire, en balayant la chaîne pointée par READERS, on lui fournira le fichier de cartes voulu. Remarquons qu'il est possible à une machine virtuelle de se faire des transferts à elle-même entre son propre perforateur et son propre lecteur.

Si cette communication entre machines peut sembler rudimentaire, c'est pourtant la seule existant officiellement dans le système standard CP distribué par le Centre Scientifique de Cambridge (il y a des mécanismes propres aux versions de CP en exploitation à Grenoble).

Les autres communications possibles n'en sont pas à proprement parler, mais sont plutôt des partages statiques d'informations :

- soit par le partage d'un disque entre toutes les machines (cas du ou des disques système) ou de disques entre plusieurs d'entre elles (pour accéder à des fichiers communs);

- soit par le partage de la mémoire (plusieurs mémoires virtuelle les peuvent être identiques ou partiellement identiques - pages partagées -). Au lieu d'avoir n copies d'un même systèm il est avantageux de réunir dans des pages que l'on partage, les parties réentrantes du système (cas de CMS).

2.2.3. Description de la communication réalisée

L'idée qui vient immédiatement à l'esprit est d'envoyer les sorties des perforateurs de toutes les machines en entrée du lecteur de la machine centralisatrice des données. Ainsi chaque machine communiquera les informations qu'elle aura recueillies en perforant les cartes.

Notons que, pour qu'il y ait transparence vis à vis de l'utilisateur, on ne peut pas immobiliser le perforateur standard d'une machine. On doit donc rajouter à la configuration de chaque machine un perforateur spécial qui, il faut le remarquer, sera toujours actif, donc aura toujours un "tampon" associé pour les E/S disques. Ce mécanisme , avec les tables qu'il implique, prend environ 50 K octets en mémoire centrale réelle du 360/67 (plus d'un K par machine virtuelle).

De plus, si l'on veut que les informations soient transmises dans l'ordre chronologique à la machine centralisatrice, il faut "fermer" les fichiers de spooling après la perforation de chaque carte, ce qui est prohibitif en temps d'exécution.

Toutes ces considérations nous ont amené à créer notre propre mécanisme de communication.

"Diagnose" spécial

Dans le jeu d'instructions d'un ordinateur IBM 360, l'instruction "diagnose" sert pour la "maintenance" à des tests "hardware" et n'est utilisée pour aucun des systèmes IBM 360 standard. CP ne simule pas cette instruction pour une machine virtuelle, ce qui n'aurait pas de sens, mais s'en sert pour permettre aux machines virtuelles d'appeler certaines de ses fonctions (un appel superviseur à haut niveau).C'est par le biais de cette instruction "diagnose" que nous allons permettre aux machines virtuelles de prévenir CP qu'elles ont de l'information à transmettre.(On reprend là une idée de Robert ADAIR de Cambridge).

Une machine virtuelle réunit les informations qu'elle beut transmettre dans une zone de 80 octets (comme pour perforer une carte), puis elle émet un "diagnose".

Le mécanisme de fonctionnement est alors le suivant :

- les informations sont transférées dans un "tampon" lié au perforateur de la machine centralisatrice des données (cette machine a été baptisée CMSMESUR), ce qui revient à simuler la perforation d'une carte sur ce perforateur ;

- si le "tampon" est plein, il est écrit sur le disque de spooling et les "tampons" sont chaînés entre eux de manière à être compatibles avec le mécanisme général de spooling de CP;

- comme d'autres "cartes" peuvent arriver durant l'écriture d'un "tampon" sur disque, il y a éventuellement constitution d'une chaîne de "tampons";

- ces fichiers de "cartes perforées" sont périodiquement "fermés" (au sens de CP) pour se prémunir contre les "cassures" du système et pour éviter d'encombrer inconsidérément le disque de spooling (les "fermetures" ont lieu toutes les 1000 cartes) ;

- ces fichiers "fermés" sont chaînés en entrée du lecteur de CMSMESUR.

- dans CMSMESUR un programme essaie en permanence de lire des cartes (s'il y a des cartes dans le lecteur virtuel, elles sont lues, sinon on se met en attente d'interruption du lecteur (cette interruption sera simulée lorsqu'on chaînera un nouveau fichier "fermé" en entrée du lecteur) pour lire à nouveau lorsque celle-ci surviendra;

- seule CMSMESUR peut déclencher la prise des mesures (mise ON du bit AUTORISE avant d'entrer dans la boucle de lecture sur le lecteur);

- la prise des mesures peut être arrêtée (mise OFF du bit AUTORISE)
  - . soit par CMSMESUR (sur interruption externe)
  - . soit à l'arrêt du système par SHUTDOWN (la commande SHUTDOWN a été modifiée pour qu'il y ait préservation des mesures faites avant l'arrêt du système).

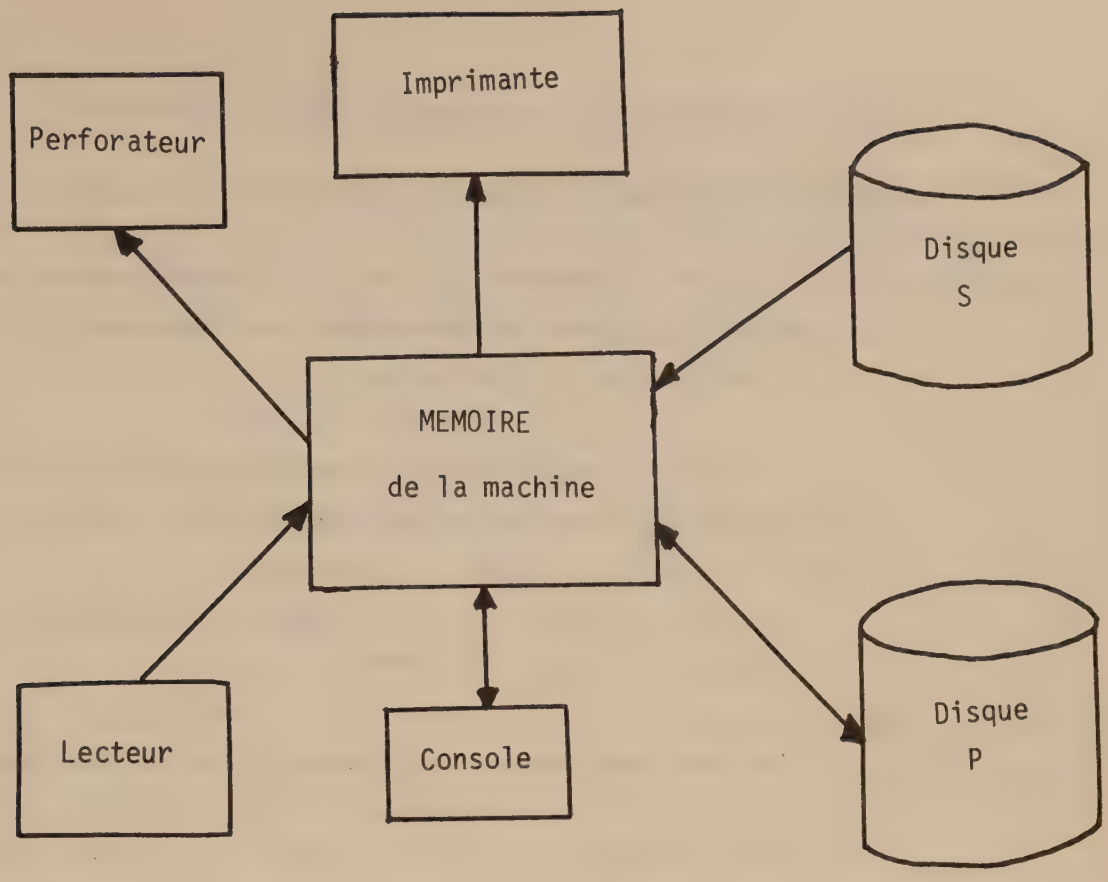
2.3. L'outil au niveau de CMS

2.3.1. Présentation de CMS [18]

Le système CMS (Cambridge Monitor System) donne à un utilisateur la possibilité d'employer, en mode conversationnel, un ordinateur IBM 360 standard (donc, en particulier, une machine virtuelle générée par le système CP 67).

L'inconvénient de cette méthode, lorsqu'on active CMS sur un 360 réel, est d'immobiliser toute une machine pour satisfaire un utilisateur; mais cet inconvénient disparaît si on utilise une machine virtuelle générée par CP.

CMS réalise la communication homme-machine par l'intermédiaire du pupitre du calculateur et de la console opérateur (s'il s'agit d'une machine virtuelle, CP simule toutes les fonctions disponibles au pupitre d'une vraie machine: touches IPL, STOP, DEPART, les clés et les voyants pour accéder aux contenus des registres ou de la mémoire) et met à la disposition de l'utilisateur la configuration minimale suivante :



- Le disque S est le disque CMS qui contient :

  . le noyau résident du système CMS qui est amené en mémoire lors de l'IPL;
  . les modules non résidents de CMS qui sont amenés en mémoire suivant les besoins au cours d'une session.

  Sous CP, le disque S est partagé entre toutes les machines.

- Le disque P est le disque réservé à l'utilisateur pour stocker ses fichiers.

- Une machine CMS (ordinateur sur lequel est activé le système CMS) peut avoir jusqu'à 6 disque : S (système),P, A, B, C, T.

Comme une machine CMS ne sert qu'un seul utilisateur, le jeu des instructions n'est pas limité. Ainsi cette possibilité d'exécuter même les instructions privilégiées permet d'utiliser CMS comme aide au développement d'un nouveau système.

Configuration mémoire d'une machine CMS

La mémoire d'une machine CMS est divisée en cinq parties :

- le noyau du système qui est constitué de la partie résidente de CMS (il gère en particulier les adresses basses ayant une signification technologique)

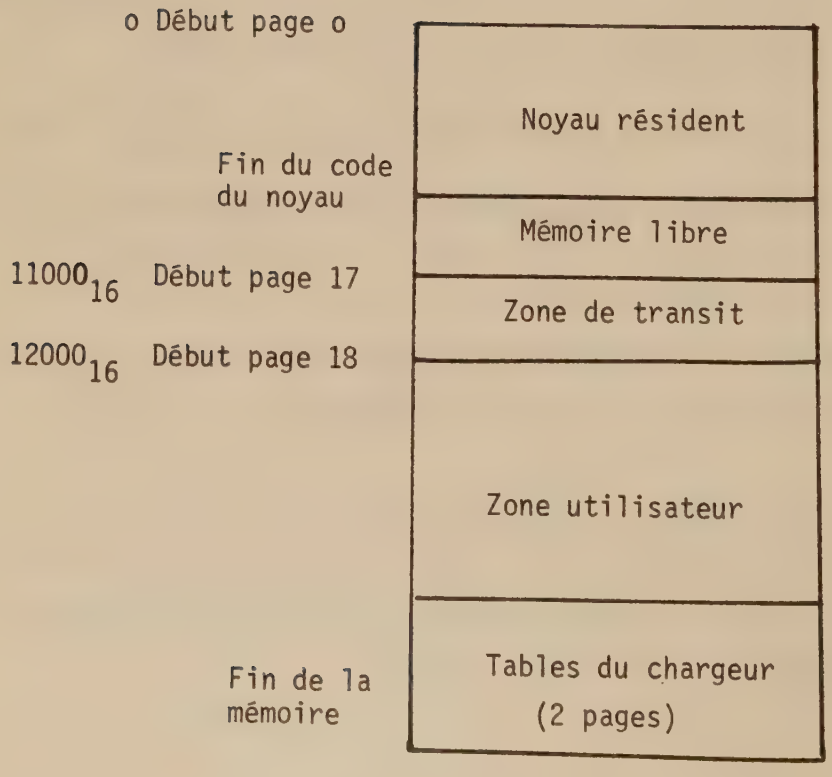
- la zone de mémoire libre du noyau (cette zone déborde, en cas de besoin, dans la partie haute de la zone utilisateur)

- la zone de transit où sont chargés les modules des commandes résidentes disque lorsqu'on en a besoin

- la zone réservée à l'utilisateur pour ses programmes

- les 8 derniers K octets (2 pages) de la mémoire sont réservés pour les tables du chargeur (résolution des problèmes d'adresses externes).

Mémoire d'une machine CMS



Commandes de CMS

L'utilisateur d'une machine CMS dispose de 75 commandes lui permettant, à partir de son terminal (pupitre de sa machine virtuelle), de demander des services au système ou de diriger ses travaux. On peut regrouper ces commandes dans les classes suivantes :

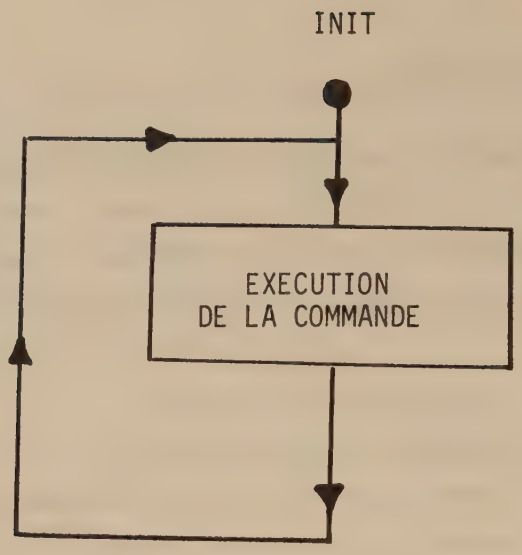
- manipulation de fichiers
- éditions de fichiers
- exécutions et mises au point de programmes
- assemblages et compilations de programmes
- divers

Ces commandes, en plus de fonctions typiquement conversationnelles, jouent un rôle analogue à celui des "cartes de contrôle" dans un système "batch" classique.

A chaque commande de CMS correspond un module système; pour les commandes les plus utilisées, le module est résident dans le noyau, pour les autres il est sur disque.

La continuité du système est assurée par le module du noyau INIT. Lorsqu'un utilisateur frappe une commande à son terminal, celle-ci est prise en compte par INIT, qui va :

- mettre la ligne utilisateur sous un format standard
- décoder la commande pour résoudre les problèmes d'abréviation éventuelle et d'utilisation de commandes définies par l'utilisateur lui-même
- donner le contrôle au module associé à la commande, après l'avoir chargé en zone de transit, si le module est résident sur disque
- remettre le système en attente de commande après l'exécution du module qui a pris le contrôle.



Commandes définies par l'utilisateur - macro-commandes

Un utilisateur a la possibilité de se définir ses propres commandes. Cette possibilité, bien que plus puissante,car elle permet l'imbrication et la récursivité des commandes définies, est analogue à celle fournie par les "procédures cataloguées" du Job Control Language d'OS.

Si lors du décodage d'une commande, on s'aperçoit (au niveau d'INIT) que cette commande n'est pas une commande primitive de CMS, on recherche parmi les fichiers de l'utilisateur un fichier ayant le nom de la commande entrée et un type spécial (type EXEC). Ce fichier, s'il existe peut être constitué: de plusieurs commandes primitives CMS, d'autres commandes utilisateur (imbrications), de la commande elle-même (récursivité) et de mots-clé spéciaux permettant :

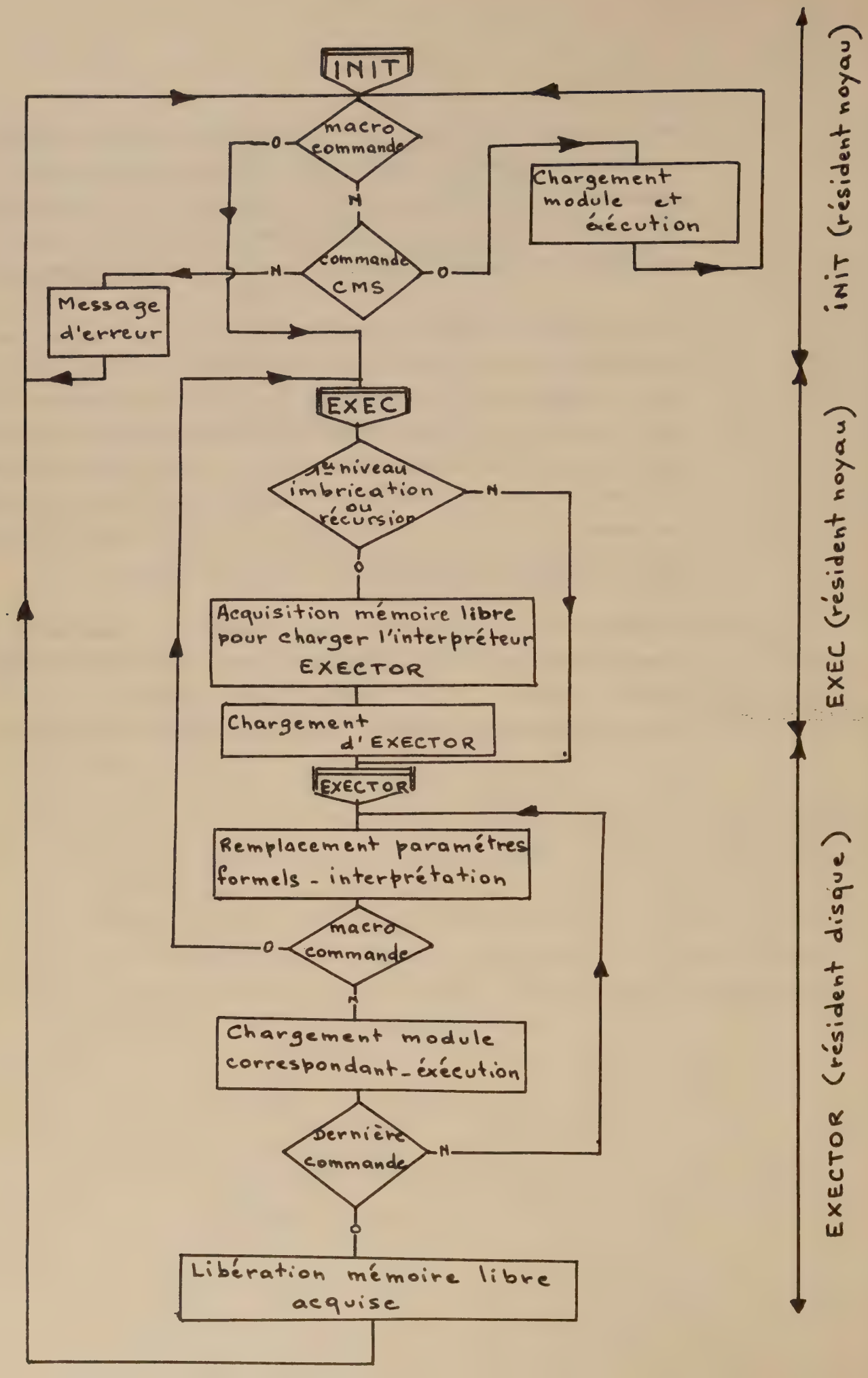
- le passage de paramètres
- des branchements (conditionnels ou non)
- la création de piles et de files d'attente
- la création de variables internes à la commande ou globales aux niveaux de récursion.

Par ce moyen, on peut, sous le nom d'une même commande, regrouper un ensemble de commandes CMS ou même faire générer un ensemble de commandes CMS (en utilisant les boucles possibles, les imbrications et la récursivité). Devant la similitude avec ce qui se passe pour la définition de macro-instructions, on parle de macro-commandes.

L'exploitation de telles macro-commandes nécessite l'emploi d'un interpréteur qui est un module (EXECTOR) résident sur disque et qui est amené en mémoire par une méthode de "boot strap" à partir du module EXEC résident dans le noyau. EXECTOR est chargé en mémoire dans une zone de mémoire libre (pratiquement dans la partie haute de la zone utilisateur), car comme une macro-commande peut contenir des commandes résidentes disque, il ne faut pas se servir de la zone de transit pour qu'il n'y ait pas écrasement.

L'imbrication et la récursivité sont assurées par l'acquisition dynamique de zones de travail en mémoire libre (1 K à chaque niveau).

Le fonctionnement à partir de l'entrée dans INIT est le suivant :


INIT
macro commande
O
N
commande CMS
N
O
Chargement module et éxécution
Message d'erreur
EXEC
1er niveau imbrication ou récursion
N
O
Acquisition mémoire libre pour charger l'interpréteur EXECTOR
Chargement d'EXECTOR
EXECTOR
Remplacement paramétres formels - interprétation
macro commande
O
N
Chargement module correspondant - éxécution
Dernière commande
N
O
Libération mémoire libre acquise
INIT (résident noyau)
EXEC (résident noyau)
EXECTOR (résident disque)

Problèmes d'adressage

Les modules du système résidant sur disque sont en binaire non translatable (image mémoire obtenue après leur chargement à une adresse précise - ici 11000_{16} ou 12000_{16} pour EXECTOR).

Le code de ces modules doit donc, à priori (nous verrons, avec EXECTOR, qu'on peut écrire du code translatable), être chargé toujours au même endroit, ici en 11000_{16}, c'est-à-dire dans la zone de transit.

Comme le noyau du système et les modules résidents sur disque sont assemblés, chargés et générés indépendamment, il se pose le problème de permettre aux modules résidents sur disque d'accéder à des variables du noyau ou d'utiliser des sous-programmes du noyau.

Ce problème d'adressage est résolu par l'implantation en mémoire basse de CMS (dans le noyau) d'un vecteur des constantes (NUCON) utilisées par les différentes routines. Ce vecteur contient en particulier une liste des adresses des constantes et sous-programmes du noyau (SYSREF).

Fichiers CMS

Les fichiers d'un utilisateur d'une machine CMS sont rangés sur les disques associés à cette machine. Chaque fichier est identifié par un triplet

<nom du fichier> <type du fichier> <mode du fichier>

Nom du fichier : chaîne quelconque de 8 caractères au plus. La chaîne doit commencer par une lettre;

Type du fichier : chaîne quelconque de 8 caractères au plus. Certains types sont réservés pour indiquer le contenu du fichier à certains processeurs du système.

FORTRAN : fichier contenant du "source" FORTRAN

SYSIN : fichier contenant du "source" assembleur 360

PLI : fichier contenant du "source" PL/1

Mode du fichier : chaîne de 2 caractères. Le premier est P, A, B, C, ou T. Le deuxième est un chiffre de 1 à 6.

La lettre indique sur quel disque se trouve le fichier et le chiffre La possibilité d'accès à ce fichier :

P1 fichier sur disque P, possibilité de lire et d'écrire

P2 fichier sur disque P, lecture seulement.

Sur chaque disque on a une table (MFD - Master File Directory) qui décrit l'organisation et l'occupation du disque, chaque fichier étant lui-même décrit par une table (FST - File Status Table). Toutes ces tables résident sur le disque concerné, mais sont emmenées en mémoire lorsque le disque est activé sur la machine CMS (commande LOGIN). Lors du fonctionnement de la machine CMS, ces tables sont entretenues en mémoire et régulièrement mises à jour sur disque (après chaque commande CMS) pour se prémunir des cassures du système.

2.3.2. Prise de mesures dans CMS

Les mesures recherchées se situant à un niveau symbolique, on va s'intéresser aux commandes CMS émises par un utilisateur pour connaître les processeurs utilisés et les ressources mises en jeu (dans notre modèle, descente des couches extérieures vers les couches internes).

Pour chaque commande, on extrait les informations suivantes :

- le nom de la machine virtuelle concernée (indicatif du 360 virtuel)
- le nom de la commande (indique le processeur mis en oeuvre: assembleur, etc...)
- s'il y a lieu, le nom du fichier concerné par la commande (par exemple, on demande de compiler un fichier TOTO)

. s'il y a lieu, le type du fichier concerné

. la taille du fichier concerné

. une indication pour savoir si le fichier a été créé ou simplement modifié par la commande employée (cas, en particulier de la commande EDIT)

. le temps virtuel mis pour exécuter la commande (temps d'exécution du 360 virtuel)

. le temps virtuel + le temps de gestion par CP pour satisfaire la commande (ce temps sert à la comptabilité, car il est proche du temps d'occupation de l'unité centrale réelle)

. le temps réel pour l'utilisateur devant sa console, entre le moment où il a frappé sa commande et celui où elle s'est terminée (on peut, par exemple, avoir une compilation PL/1 de 50 secondes dans le 360 virtuel, de l'ordre de 70 secondes en unité centrale réelle, mais pour l'utilisateur devant son terminal, elle a duré 20 minutes!).

. l'heure d'émission de la commande (donc à un moment plus ou moins chargé de la journée)

. une indication pour savoir si la commande était erronnée ou non (commande inexistante, faute de compilation, etc...)

. le nombre de pages en mémoire, ce qui donne approximativement le "working set" de la commande

. le nombre de fautes de pages qui ont été nécessaires (interruption dûe à une page non en mémoire)

. le nombre d'entrée-sortie disque qui ont été nécessaires

. le nombre d'entrée-sortie console qui ont été nécessaires

. le nombre d'utilisateurs du système au moment de l'émission de la commande (système plus ou moins chargé)

. le niveau de récursion ou d'imbrication d'une commande (0 pour une commande primitive CMS entrée au terminal, 1 pour une commande provenant d'un fichier EXEC au premier niveau de récursion ou d'imbrication, etc...)

Pour pouvoir prendre ces mesures, en particulier celles relatives au temps, il faut connaître à la fois le début et la fin d'une commande. C'est pourquoi des modifications ont été apportées à INIT, dans le noyau, pour les commandes entrées directement au terminal et à l'interpréteur EXECTOR pour les commandes provenant d'un fichier EXEC (macro-commandes). Les modifications du module EXECTOR ont entraîné celles du module EXEC du noyau qui emmène EXECTOR en mémoire : il y a passage d'un vecteur d'adresses pour qu'EXECTOR connaisse l'emplacement dans le noyau de certains compteurs (compteurs d'entrée-sortie) et de certains sous-programmes.

L'essentiel des modifications faites à INIT et EXECTOR tient dans :

. l'adjonction d'un petit décodeur de commandes.
Il faut en effet décoder les commandes pour connaître la signification des paramètres passés et connaître, en cas de fichiers, certains types implicites.

. une recherche dans les tables décrivant les fichiers de l'utilisateur

. un calcul des différents temps (en se servant de l'horloge externe simulée par CP - unité d'adresse OFF -)

. un relevé des divers compteurs.

Les modules DIO (entrée-sortie disque) et CONSI (entrée-sortie console) du noyau ont été modifiés pour entretenir des compteurs qui ont été rassemblés dans INIT (DIO et CONSI étant réentrants et de plus DIO se trouvant dans les pages partagées de CMS).

Il faut parler de la manière particulière dont est écrit le code d'EXECTOR (et donc les modifications qui y ont été apportées).

EXECTOR est un module qui a été généré après chargement à une adresse précise de la mémoire, mais l'image mémoire obtenue doit être chargeable n'importe où en mémoire, car l'emplacement où on va charger EXECTOR est obtenu par un mécanisme de gestion de mémoire libre. Ceci implique:

- que toutes les adresses sont exprimées sous la forme base + déplacement. Les bases étant chargées par EXEC lors de l'acquisition de la zone de mémoire libre;

- que s'il n'est pas possible de mettre les adresses sous forme base + déplacement (cas des appels par SVC avec passage de liste de paramètres), ces dernières doivent être calculées dynamiquement avant leur utilisation.

2.4. Fonctionnement général

2.4.1. La machine CMSMESUR

La machine virtuelle, qui a pour indicatif CMSMESUR, centralise les données qui lui sont envoyées dans son lecteur de cartes. Lorsque le mécanisme de prise de données n'est pas activé, CMSMESUR n'a pas de rôle privilégié par rapport aux autres machines virtuelles.

Pour activer le mécanisme de prise de données, il faut lancer en exécution dans CMSMESUR un programme spécial, qui :

- sauve éventuellement sur bande magnétique les mesures déjà faites,

- si on ne sauve pas sur bande, change les noms des fichiers des mesures déjà existants par sécurité,

. émet un diagnose avec un paramètre spécial permettant de déclencher les mesures,

. se met en attente de lecture sur le lecteur de cartes

. à chaque lecture d'un fichier de cartes (1000 cartes) le met sur disque.

Les fichiers de mesures sur disque (disque P de la machine CMSMESUR) sont des fichiers CMS classiques. Ils sont identifiés par des noms réservés et des types constitués de la date de la prise des mesures et d'un numéro d'ordre dans la journée (ces noms et ces types sont attribués automatiquement).

Cette partition des mesures en plusieurs fichiers de noms différents a été réalisée pour des raisons de sécurité, afin :

. d'éviter l'écrasement de deux fichiers ayant le même nom,

. de ne pas perdre un trop grand nombre de mesures (par exemple, une journée complète), en cas d'ennui.

2.4.2. Fonctionnement du point de vue de l'opérateur

Sauf au lancement de CP, l'opérateur du système n'a pas à s'inquiéter de la prise des mesures.

Au démarrage de CP, si on veut prendre des mesures durant la session, l'opérateur doit :

. activer la machine CMSMESUR,

. lancer sur cette machine le programme spécial "MESURE" (il suffit de frapper "MESURE" sur le terminal).

- éventuellement monter une bande magnétique sur un dérouleur de bandes, s'il a répondu "OUI" à la question "SAUVE-T-ON SUR BANDE ?" (chaque matin, sauvegarde sur bande des mesures de la journée précédente)

- déconnecter la machine CMSMESUR (la machine virtuelle continue à "tourner", mais elle n'est plus associée à une console - libération du terminal employé -).

Les mesures seront arrêtées et sauvegardées automatiquement à l'arrêt du système CP. Si on désire les arrêter en cours de session, il suffit de reconnecter la machine CMSMESUR (on lui associe à nouveau une console) et de provoquer une interruption externe sur cette machine (commande EXTERNAL).

CHAPITRE 3

<><*****><>

3. LES PREMIERS RESULTATS

3.1. Configuration et environnement de l'IMAG

Le système CP/CMS mesuré est constitué des systèmes CP-67 (Version 3.0) et CMS (version 3.0) et tourne sur la configuration suivante (cf ANNEXE 3) :

- une unité IBM 360/67
- une mémoire centrale de 1024 K octets
- 3 canaux simples :
  - . 1 relié à 2 tambours pour la pagination
  - . 2 reliés chacun à une unité de 8 disques 2314
- 1 canal multiplexeur où sont raccordés :
  - . 50 terminaux
  - . 4 dérouleurs de bandes magnétiques
  - . 2 imprimantes
  - . 2 lecteurs-perforateurs

CP tourne de 7 H à 12 H et de 15 H à 19 H, aux autres heures le système actif est OS/360 d'IBM.

La population des utilisateurs est constituée de divers chercheurs: en physique, analyse numérique, langages, systèmes informatiques, etc..., et le samedi matin, d'étudiants. Ces utilisateurs disposent de 300 machines virtuelles reconnues par CP et d'un jeu de 75 commandes primitives CMS pour diriger leurs travaux.

Deux prises de mesures ont été effectuées du 11.03 au 06.04.1972 et du 30.06 au 28.07.1972. Devant l'analogie des deux échantillons (fréquences de commandes, temps moyens, etc...cf: ANNEXE 4) et le coût des temps de calcul nécessaires, une analyse poussée des données n'a été entreprise que pour le premier échantillon. Précisons, à ce sujet, que les résultats donnés en ANNEXE 4 demandent, pour chaque échantillon, environ 1000 secondes d'unité centrale(IBM 360/67).

Par ailleurs, une analyse journée par journée montre que la stabilisation de l'utilisation est acquise au bout d'une semaine de fonctionnement.

Caractéristiques générales de l'échantillon traité.

Les résultats que nous donnons sont globaux et concernent les 60.000 commandes CMS recueillies entre 11.03 et le 06.04.1972.

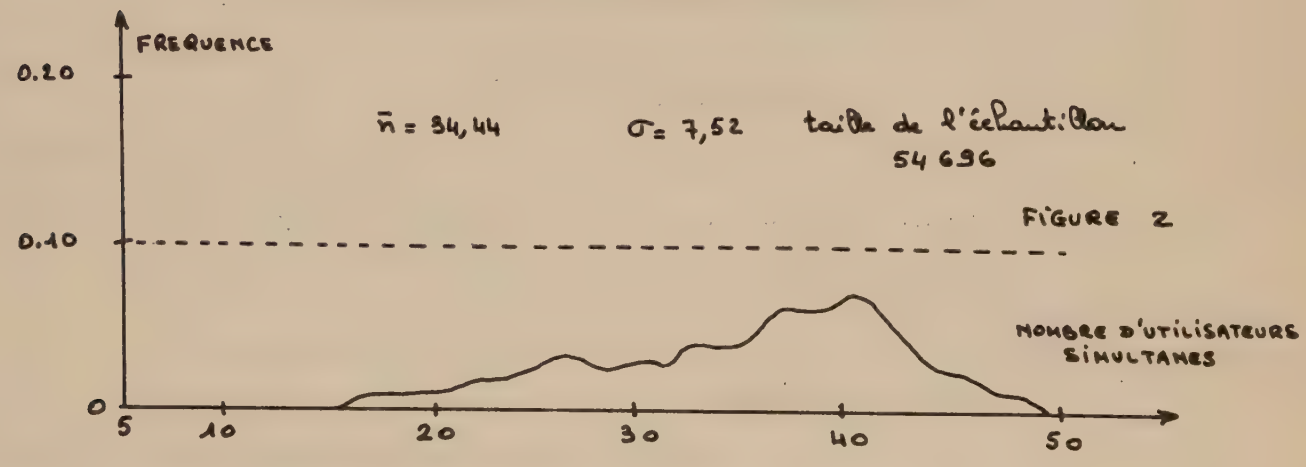
Les FIGURES 1 à 3 résument le comportement des utilisateurs :

- La FIGURE 1 permet de voir que le système est plus chargé l'après-midi que le matin.

- La FIGURE 2 donne la fréquence en fonction du nombre d'utilisateurs simultanés (approximation de la densité de probabilité). L'intégrale de cette fréquence donne la fréquence cumulée (approximation de la fonction de répartition de la loi).
  Notons que les utilisateurs se limitent d'eux-mêmes en fonction du temps de réponse (décroissance régulière de 40 à 50 utilisateurs).

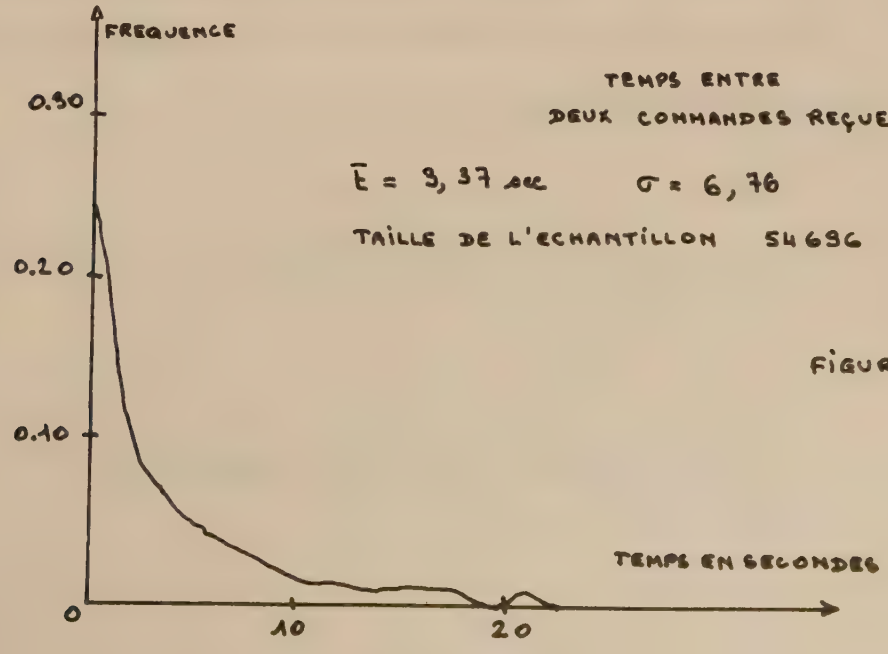
- La FIGURE 3 indique la loi suivie par le temps entre l'arrivée de 2 commandes CMS dans le système (les 2 commandes peuvent provenir de machines virtuelles différentes). Ces commandes sont des commandes logiques et ne reflètent pas l'activité des terminaux, du fait de l'existence de macro-commandes et de caractères spéciaux découpant une ligne physique en plusieurs lignes logiques. Cependant nous verrons qu'il est possible de reconstituer grossièrement l'activité moyenne d'un terminal.

FIGURE 1



$\bar{n} = 34,44$ $\sigma = 7,52$ taille de l'échantillon 54 696

FIGURE 2



TEMPS ENTRE DEUX COMMANDES REÇUES

$\bar{t} = 9,37$ sec $\sigma = 6,76$

TAILLE DE L'ECHANTILLON 54 696

FIGURE 3

Quelques grandeurs

Taille de l'échantillon	59301 commandes
Nombre de machines virtuelles activées	256 (sur 300)
Jeu de commandes CMS employées	66 (sur 75)
% de commandes inexistantes (erreur)	2,21 %
% de commandes retournant un code d'erreur	11,71 %
% des commandes abrégées (O pour OFFLINE, F pour FORTRAN) etc...	29,01 %
% d'appels de macro-commandes	7,23 %
Niveau maximum de récursion atteint	6

3.2. Fréquences - Moyennes - Histogrammes

3.2.1. Généralités [24]

Définissons tout d'abord les premiers outils statistiques qui nous ont permis de réduire l'ensemble des données recueillies et d'en commencer l'interprétation.

Valeurs centrales

Dans un premier stade de réduction, les variables mesurées peuvent être ramenées à des valeurs centrales telles que :

a - Moyenne

$\overline{X}$ = moyenne de la variable X mesurée par les n valeurs x_i

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Il faut noter que l'expression $\sum_{i=1}^{n} (x_i - a)^2$ est minimum pour $a = \overline{X}$

Dans le cas où X est une loi de probabilité de densité f, on définit l'espérance mathématique de la variable X :

$$E(X) = \int_{-\infty}^{+\infty} t \quad f(t)dt$$

On dit que $\overline{X}$ est un estimateur de E(X).

b - *Médiane*

$\tilde{X}$ = médiane de la variable aléatoire X de densité f

$$\int_{-\infty}^{\tilde{X}} f(t)dt = \frac{1}{2}$$

Autre notation avec F, fonction de répartition de X :

$$\tilde{X} = F^{-1}\left(\frac{1}{2}\right)$$

Mesures de dispersion

S'il est intéressant de connaître les valeurs centrales d'une variable qu'on étudie, il l'est aussi de savoir comment la variable se comporte autour de ces valeurs.

a - *Etendue*

L'étendue de la variable X mesurée par les valeurs x_i est :

$$\max_{i=1,n} x_i - \min_{i=1,n} x_i$$

L'étendue de X indique le domaine dans lequel la variable a été mesurée.

b - *Ecart moyen*

$$E.M. = \frac{1}{n} \sum_{i=1}^{n} |x_i - \overline{X}|$$

c - Ecart-type

En probabilité, on définit l'écart-type d'une variable aléatoire X, comme

$$\sigma(X) = \sqrt{E((X-E(X))2)}$$

Ce qui peut s'écrire en développant :

$$\sigma(X) = \sqrt{E(X^2) - E(X)^2}$$

Statistiquement , $\sigma(X)$ peut être estimé par

$$s(X) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{X})^2} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2 - (\frac{1}{n} \sum_{i=1)}^{n} x_i)^2}$$

La seconde forme de s(X) est préférable, car elle est calculable en un seul balayage des valeurs x_i .

Il faut remarquer que l'expression

$$\sum_{i=1}^{n} (x_i - \bar{X})^2$$

est minimum du fait de la propriété de la moyenne.

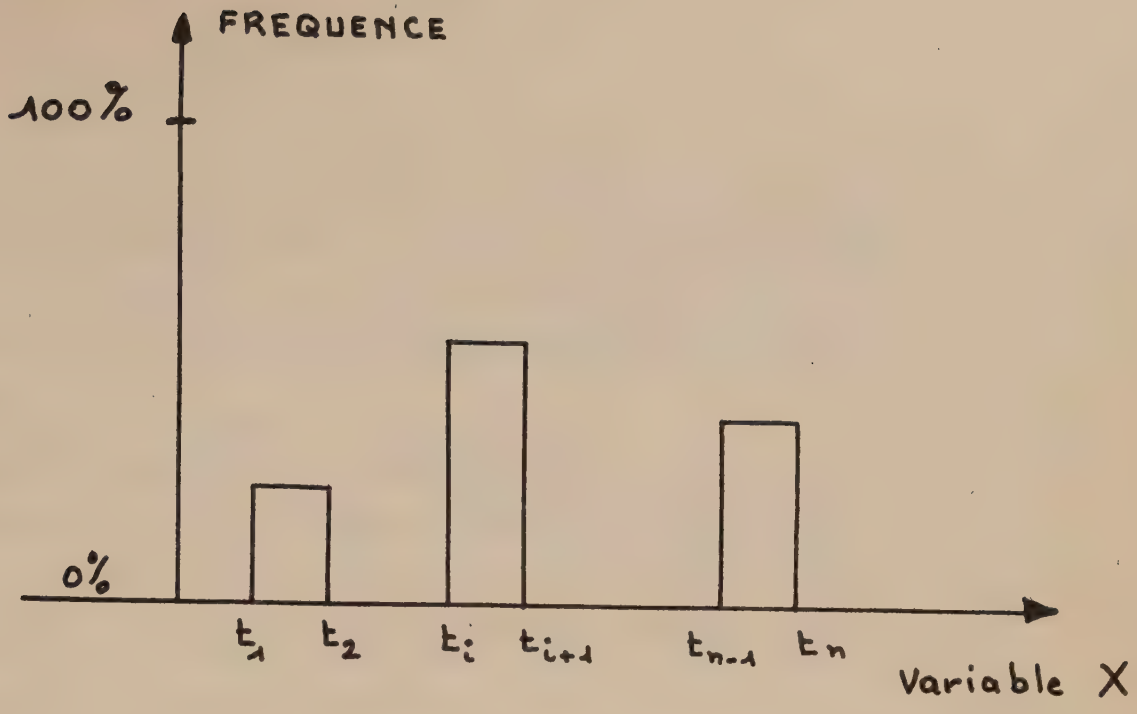
Finalement on montre qu'un estimateur sans biais de $\sigma(X)$ est donné par

$$\sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{X})^2}$$

Histogrammes

Généralement une moyenne et un écart-type ne suffisent pas à caractériser une variable, il est intéressant d'avoir sa densité de probabilité.

Statistiquement, cette densité peut être approchée par des histogrammes où on représente graphiquement la fréquence des valeurs de la variable X qui sont dans l'intervalle $[t_i, t_{i+1}[$



Pour que l'histogramme approche bien la densité de probabilité, il faut que le pas d'échantillonnage $t_{i+1} - t_i$ soit judicieusement choisi.

Dans les résultats que nous donnons les pas sont le 1/10e des graduations sur les abscisses (exemple : pour les E/S disque le pas est de 40) et ils ont été déterminés par tatonnements après plusieurs essais.

REMARQUE

La médiane représente :
- dans un histogramme simple :
  l'axe qui partage l'histogramme en 2 aires égales
- dans un histogramme cumulé
  l'abscisse du point qui a pour fréquence 50 %.

3.2.2. Résultats pour CP/CMS

Pour chaque commande on a calculé sa fréquence, le

le nombre moyen de ses E/S disque et console, la taille moyenne des fichiers manipulés et la fréquence d'apparition des codes d'erreurs. Tous ces résultats sont rassemblés dans l'ANNEXE 4 où ils sont donnés globalement pour chaque échantillon de prises de mesures.

On donne aussi des statistiques sur les manipulations de fichiers suivant leur contenu. De telles mesures peuvent intéresser les concepteurs de systèmes chargés de réaliser des méthodes d'accès. On peut remarquer qu'elles reflètent l'utilisation des composants du système (FORTRAN, ASSEMBLEUR, PL/1, etc...).

Les commandes pour lesquelles on indique une longueur de fichier nulle sont :

- soit des commandes non décodées (cf ANNEXE 1)

- soit des commandes qui n'ont pas de fichiers pour paramètres.

Nous avons regroupé des résultats de fréquences d'évènements et de temps dans un tableau (Fig.4) où les commandes sont réparties suivant les 5 catégories logiques de SCHERR [32] :

1- Manipulations de fichiers
2- Editions de fichiers
3- Chargements, mises au point et exécutions
4- Assemblages et compilations
5- Divers

Signification	Manipulation de fichiers	Edition de fichiers	Chargements Mises au point Exécution	Assemblages et Compilations	Divers
Fréquence	36,90 %	12,17 %	16,93 %	5,06 %	21,69 %
Pourcentage U.C. virtuelle (CMS)	4,53 %	5,42 %	54,73 %	33,78 %	1,54 %
Pourcentage U.C. Réelle (CP)	7,69 %	9,58 %	48,15 %	32,20 %	2,38 %
Pourcentage Temps réel pour l'utilisateur	5,05 %	51,67 %	28,55 %	11,90 %	2,82 %
Overhead	62,14 %	63,69 %	26,98 %	32,60 %	58,47 %
E/S disque moy	18,65	65,25	99,75	491,77	34,04
E/S disque σ	71,37	116,59	279,72	786,44	237,89
E/S console moy	1,45	30,00	14,07	9,32	5,47
E/S console σ	10,11	51,23	54,83	40,15	34,96
Temps virtuel en sec. moy	0,147	0,524	3,406	7,85	0,095
Temps virtuel en sec. σ	0,524	1,256	30,281	10,94	1,550

Figure 4

De ce tableau on peut tirer les remarques suivantes :

- les assemblages et les compilations représentent 1/3 de la charge de l'unité centrale, ce qui justifie le coût des optimisations qu'on peut apporter (algorithmes adaptés).

- les chargements et exécutions représentent la moitié de la charge de l'U.C., donc l'autre moitié est constituée de programmes et modules standard dont on peut étudier le comportement aux fins d'optimisation.

- l'édition de fichiers ne représente que 9,58 % de la charge, mais 50 % du temps de l'utilisateur.

La partition des commandes en 5 groupes logiques est assez arbitraire. En particulier, la commande GLOBAL qui sert à indiquer des appels éventuels de bibliothèques est très courte (simples modifications de tables). Classée dans la catégorie 3, elle a contribué, par sa fréquence importante, à raccourcir d'une manière non négligeable le temps moyen d'exécution de cette catégorie.

Si on ne retrouve pas 100 % comme somme des fréquences de commandes, c'est que les 7,23 % d'appels de macro-commandes ne sont pas comptabilisés.

REMARQUES

L'"overhead" n'est pas l'overhead habituel. C'est le travail représentant les "frais généraux" du système, qui se divise, sous CP/CMS, en 2 parties :

- la partie imputable à un utilisateur (services spéciaux, simulations d'instructions privilégiées) qui est la seule comptabilisée ici.
- la partie dûe aux mécanismes de CP (pagination, spooling) qui n'est pas imputée aux utilisateurs.

Le temps réel pris en compte est le temps réel actif (on ne tient pas compte du temps de réflexion de l'utilisateur entre 2 commandes CMS, ni du temps passé pour entrer la ligne du terminal). Pourtant des figures 2 et 3, on peut déduire qu'un utilisateur émet en moyenne une commande CMS logique toutes les 134 secondes (≃ 34,44 x 3,87).

Histogrammes

Les histogrammes des temps d'exécutions, des nombres d'E/S disque et console ont été tracés pour chaque commande. On donne dans les figures de 5 à 11 les résultats qui sont les plus intéressants; en particulier, les histogrammes relatifs à PL/1, FORTRAN IV G et ASSEMBLEUR G, "processeurs" qui représentent 28,40 % de la charge du système (sur les 32,20 % de la catégorie 4).

L'ANNEXE 6 donne des résultats complémentaires :

- sur des fréquences et pourcentages commande par commande
- sur des histogrammes associés aux commandes d'exécutions de programmes utilisateurs (pour mieux voir ce qui se passe dans la catégorie 3 des commandes).

REMARQUES :

Fortran semble très efficace comparé à PL/1 qui est responsable du second pic des E/S disque dans la catégorie 4.
Notons aussi l'importance des E/S disque faites par l'ASSEMBLEUR.
Il faut remarquer qu'à partir de ces résultats, on peut reconstituer certaines valeurs moyennes non mesurées directement.
Ainsi pour le "temps de réflexion" moyen d'un utilisateur (temps mort entre 2 commandes).

temps moyen entre 2 commandes d'un utilisateur	134 sec (fig 2 et 3 : 34,44 x 3,87)

temps moyen d'exécution d'une commande	69 sec (fig 5)
temps de réflexion moyen pour un utilisateur	65 sec (134-65)

On peut même en déduire l'activité moyenne d'un terminal.

temps moyen entre 2 E/S console pour un terminal	16,4 sec (ci-dessus et fig.5 134/8,18)

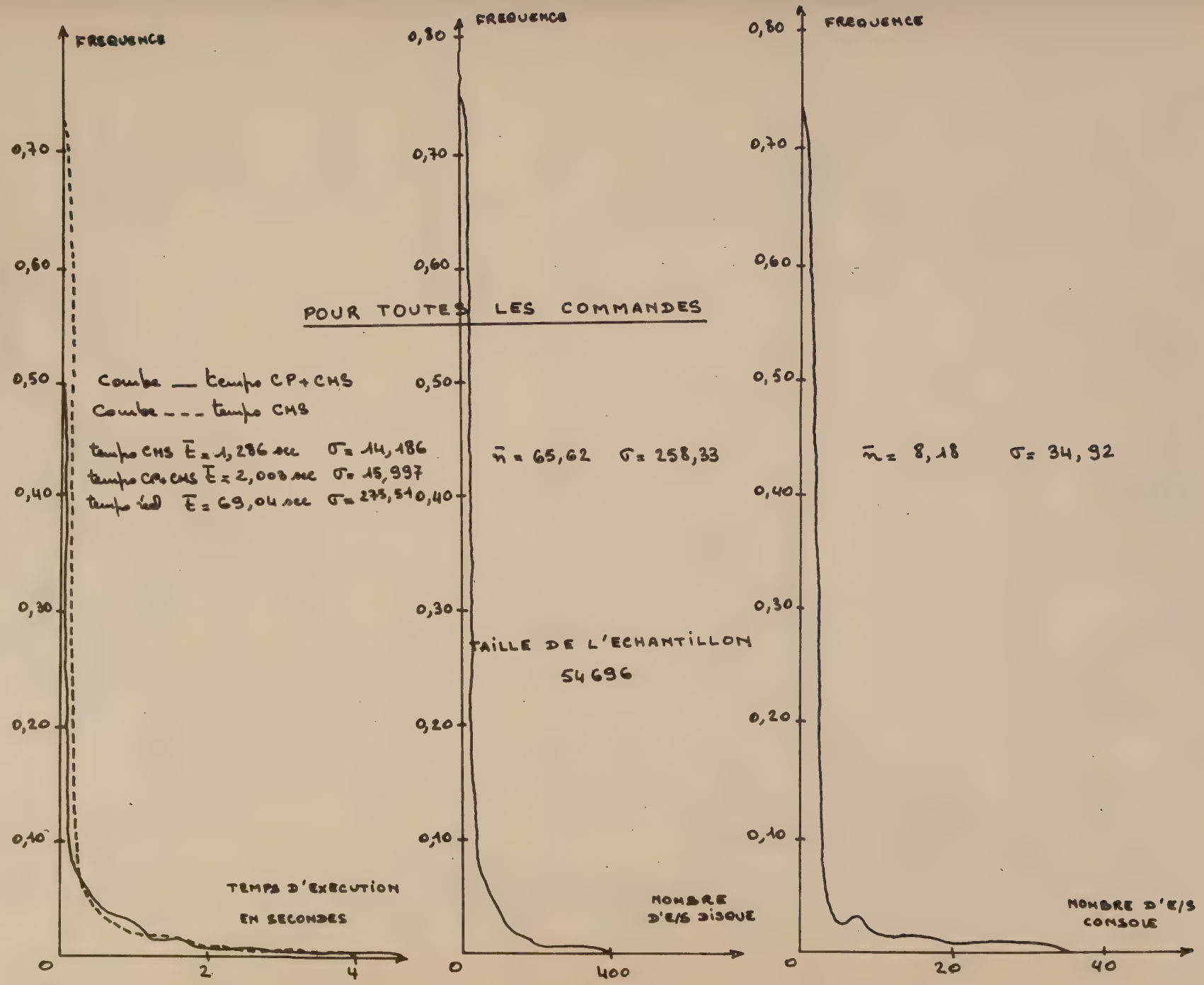
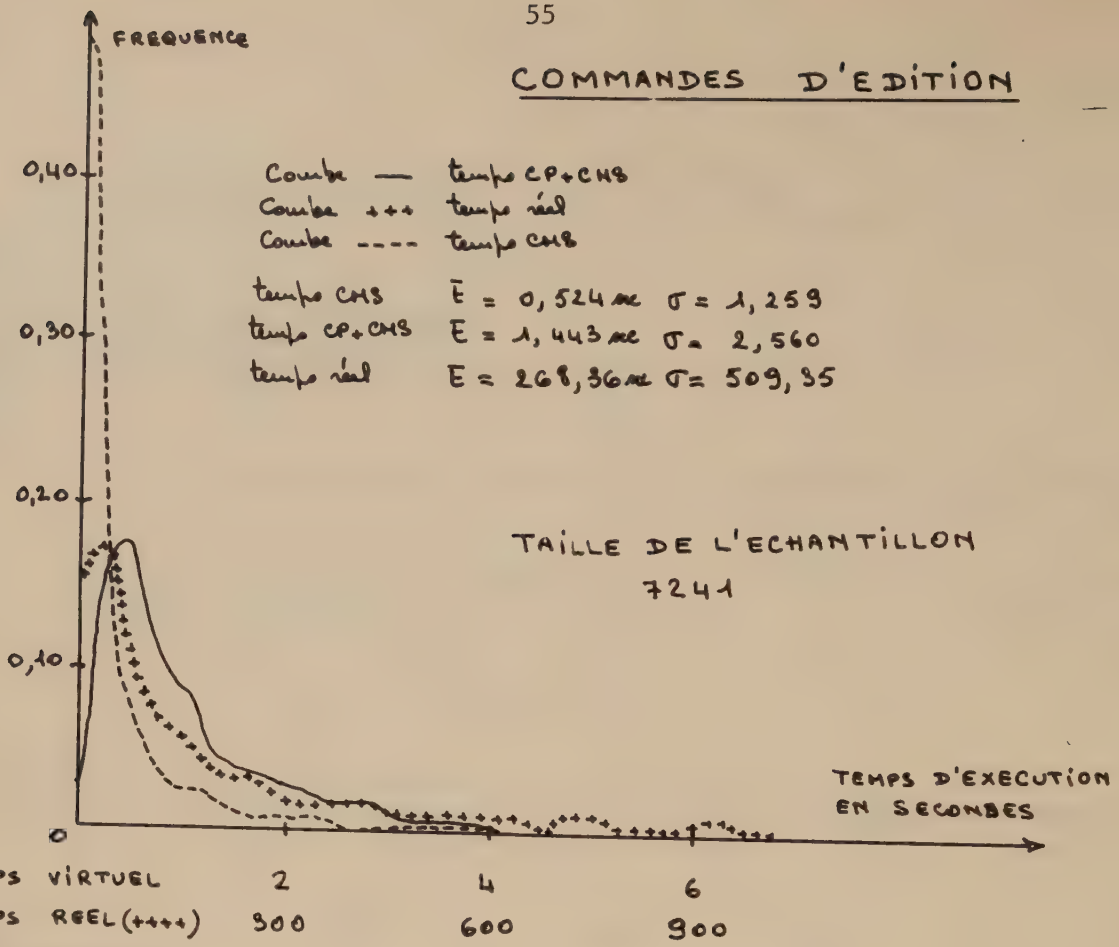

FREQUENCE
0,80
0,70
0,60
0,50
0,40
0,30
0,20
0,10
0
2
4
TEMPS D'EXECUTION EN SECONDES
Courbe ___ temps CP+CMS
Courbe - - - temps CMS
temps CMS $\bar{E}$ = 1,286 sec σ = 14,186
temps CP+CMS $\bar{E}$ = 2,008 sec σ = 15,997
temps réel $\bar{E}$ = 69,04 sec σ = 273,51
POUR TOUTES LES COMMANDES
$\bar{n}$ = 65,62 σ = 258,33
TAILLE DE L'ECHANTILLON
54696
400
NOMBRE D'E/S DISQUE
$\bar{n}$ = 8,18 σ = 34,92
20
40
NOMBRE D'E/S CONSOLE

FIGURE 5

COMMANDES D'EDITION



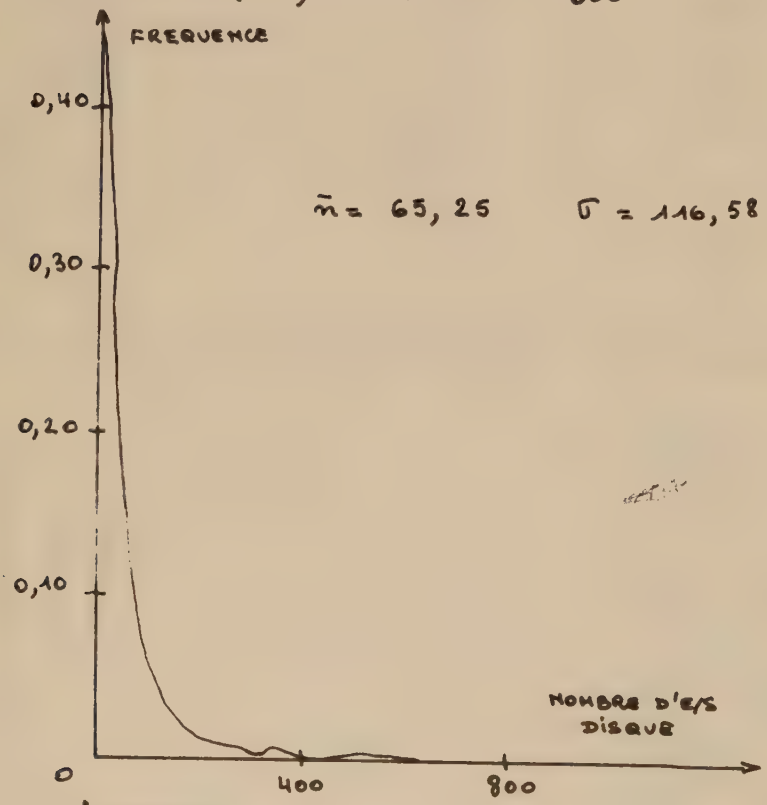
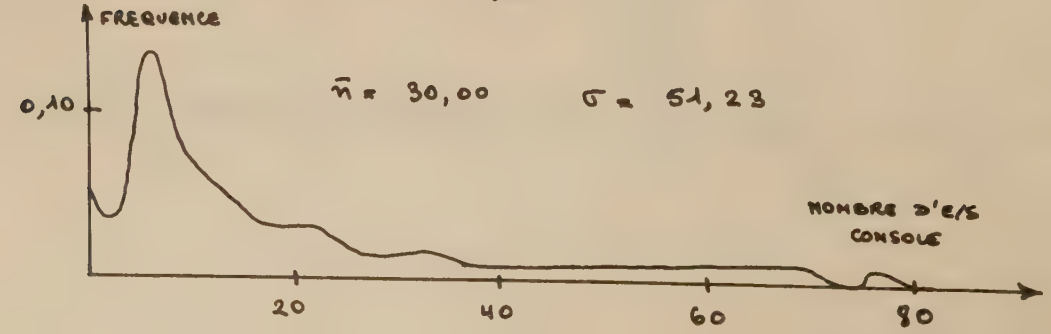
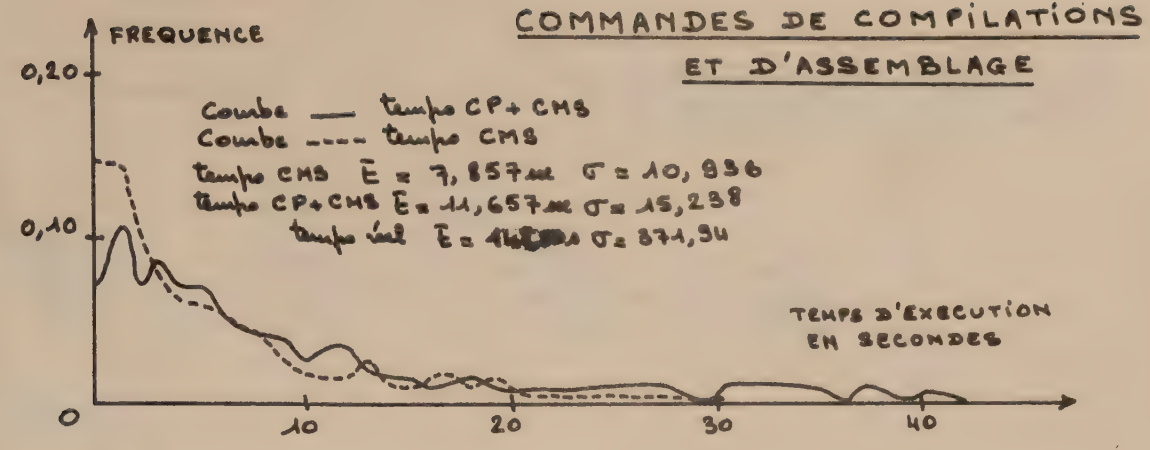


FIGURE 6





TAILLE DE L'ECHANTILLON 3026

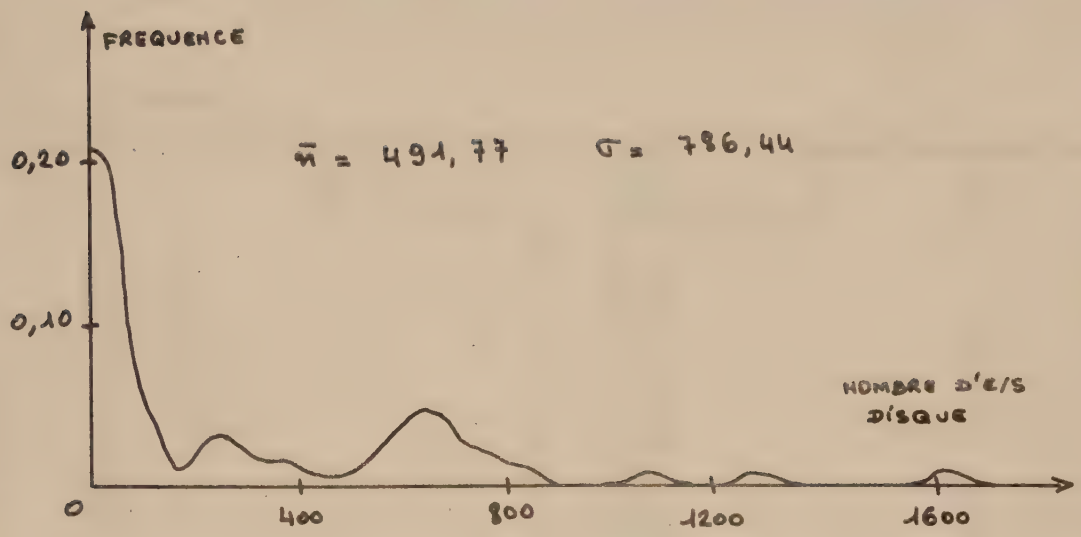
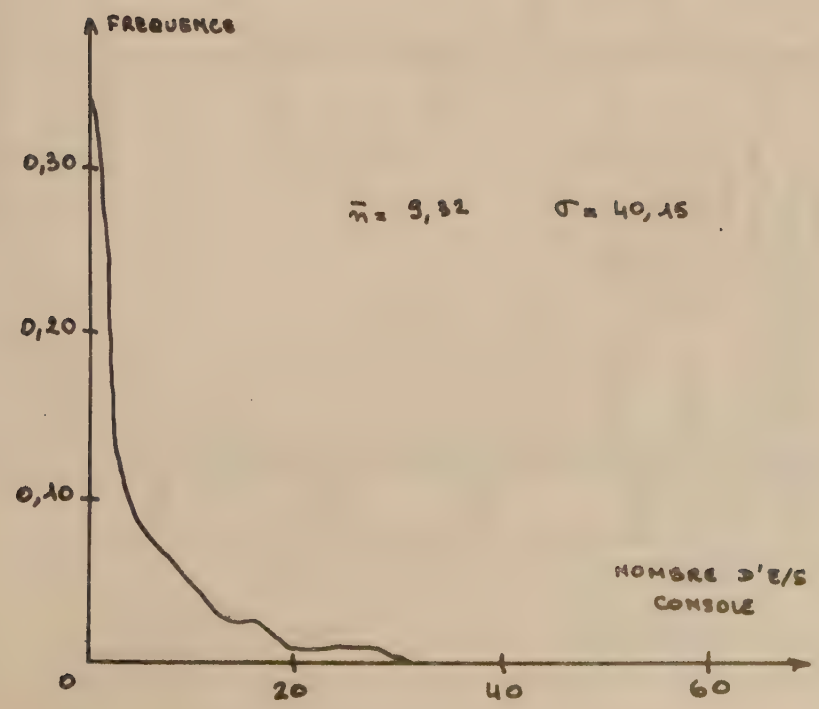


FIGURE 7



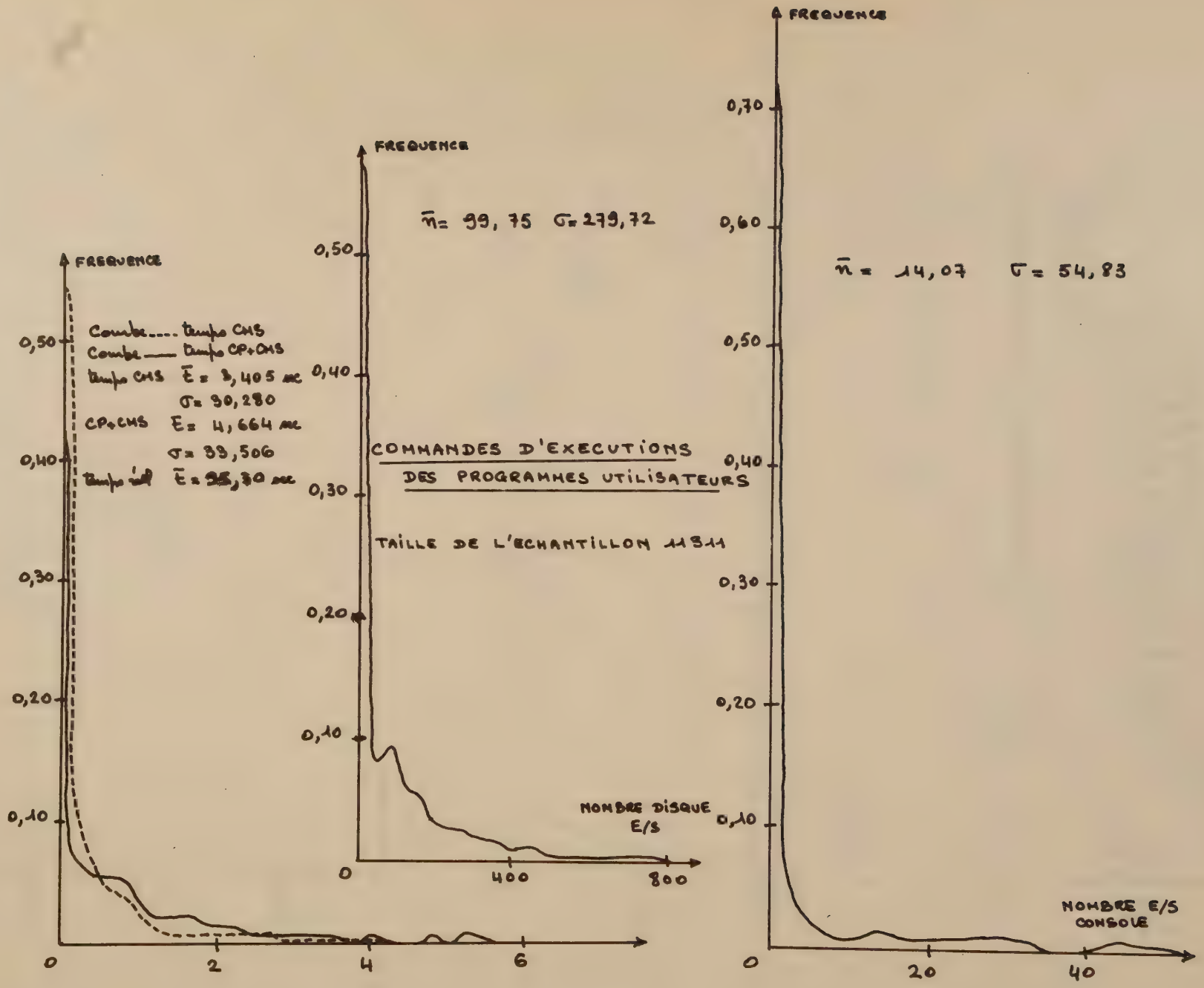
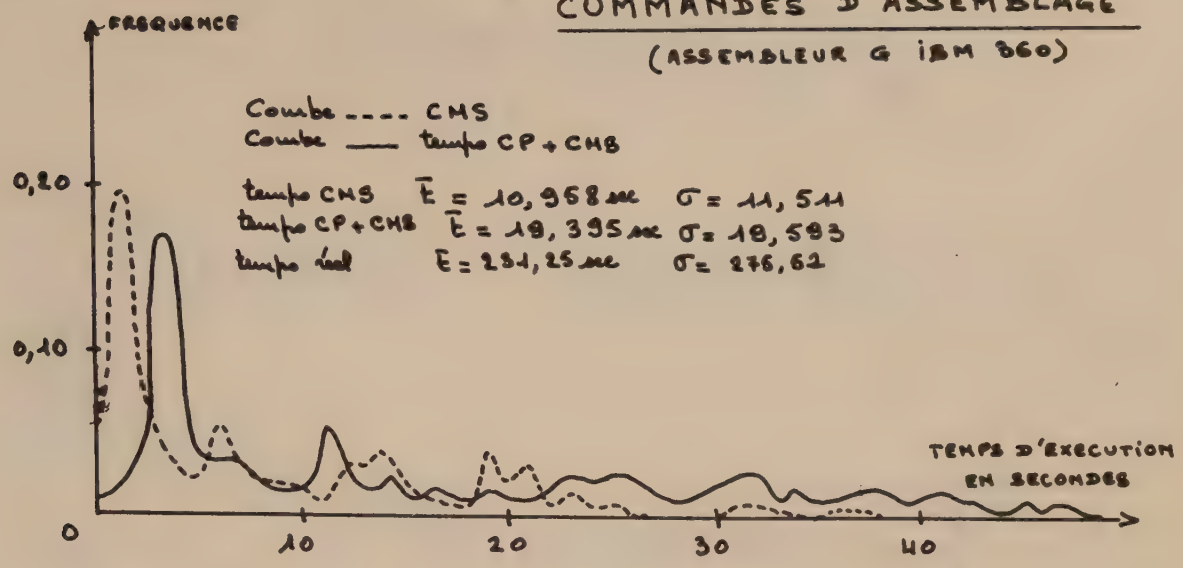
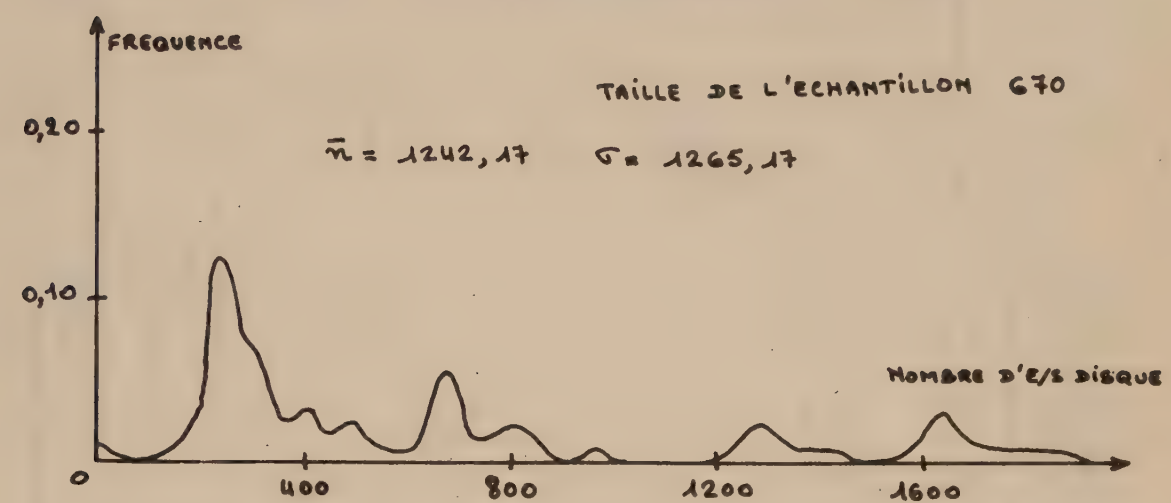

FREQUENCE
Courbe.... temps CMS
Courbe —— temps CP+CMS
temps CMS $\bar{t}$ = 3,405 sec
σ = 30,280
CP+CMS $\bar{t}$ = 4,664 sec
σ = 33,506
temps réel $\bar{t}$ = 25,30 sec
0,50
0,40
0,30
0,20
0,10
0
2
4
6
FREQUENCE
$\bar{n}$ = 99,75 σ = 279,72
COMMANDES D'EXECUTIONS
DES PROGRAMMES UTILISATEURS
TAILLE DE L'ECHANTILLON 11311
0,50
0,40
0,30
0,20
0,10
NOMBRE DISQUE E/S
0
400
800
FREQUENCE
$\bar{n}$ = 14,07 σ = 54,83
0,70
0,60
0,50
0,40
0,30
0,20
0,10
NOMBRE E/S CONSOLE
0
20
40

FIGURE 8

COMMANDES D'ASSEMBLAGE

(ASSEMBLEUR G IBM 360)





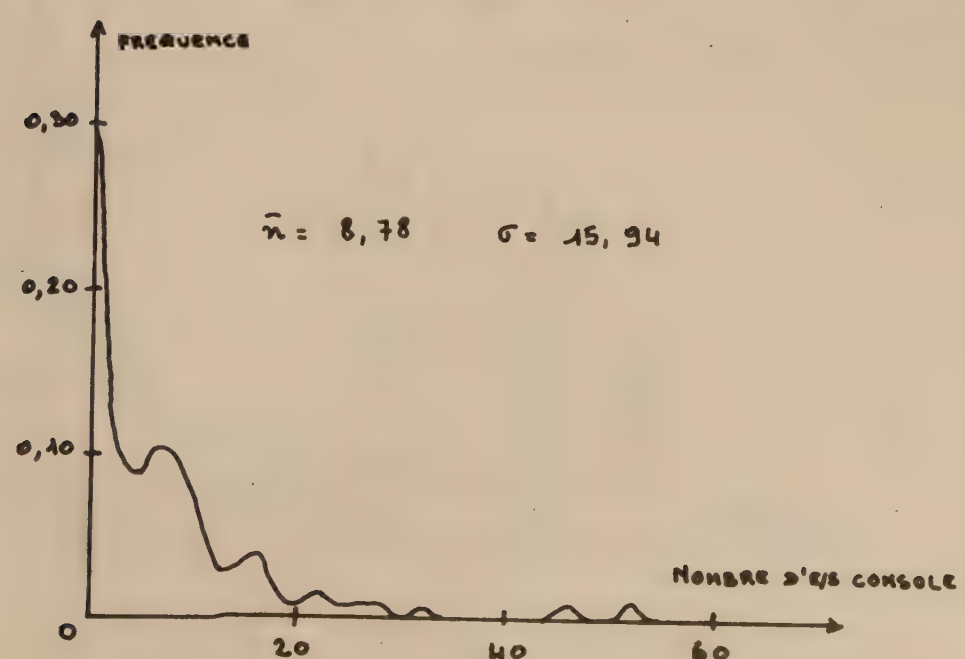
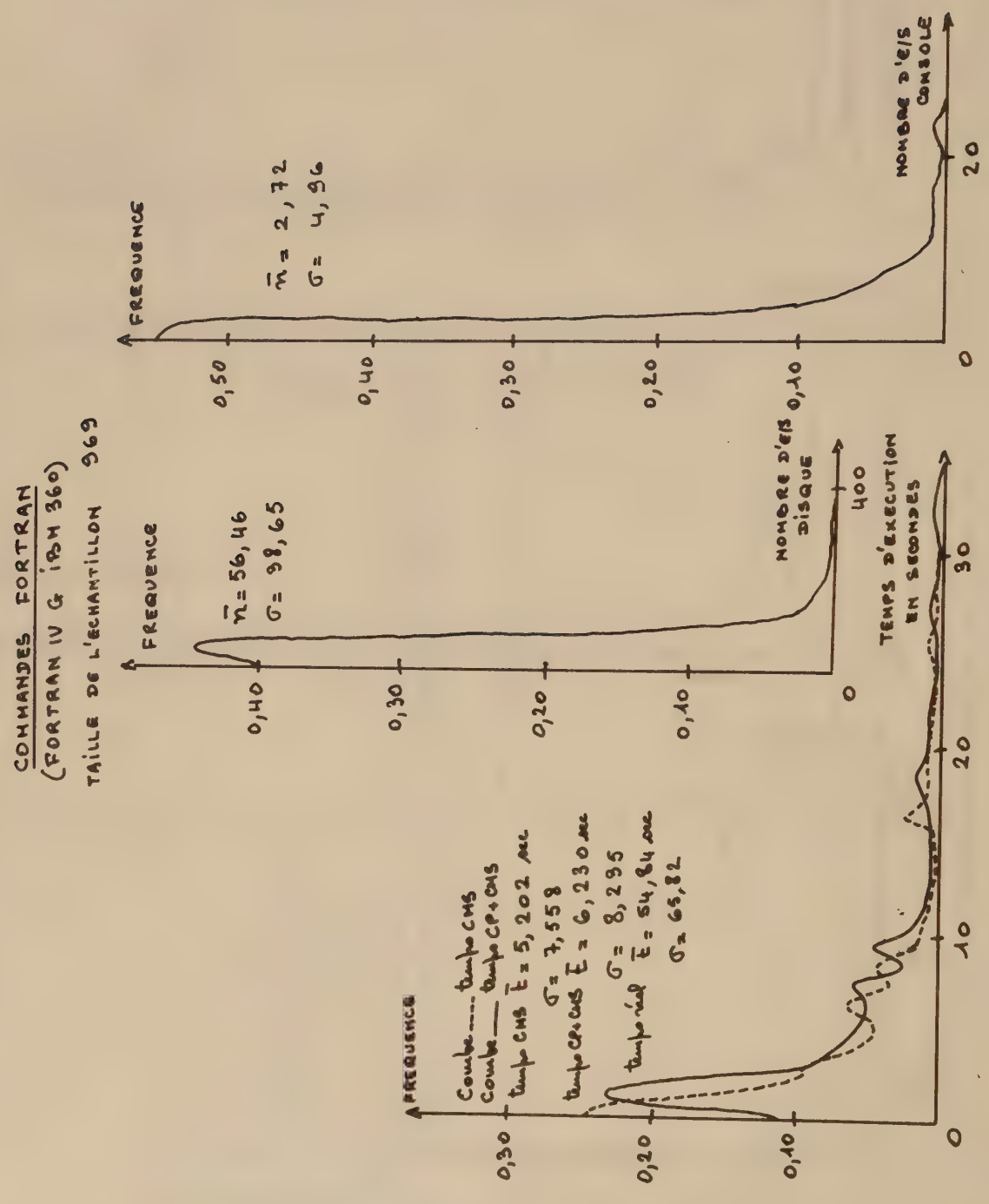


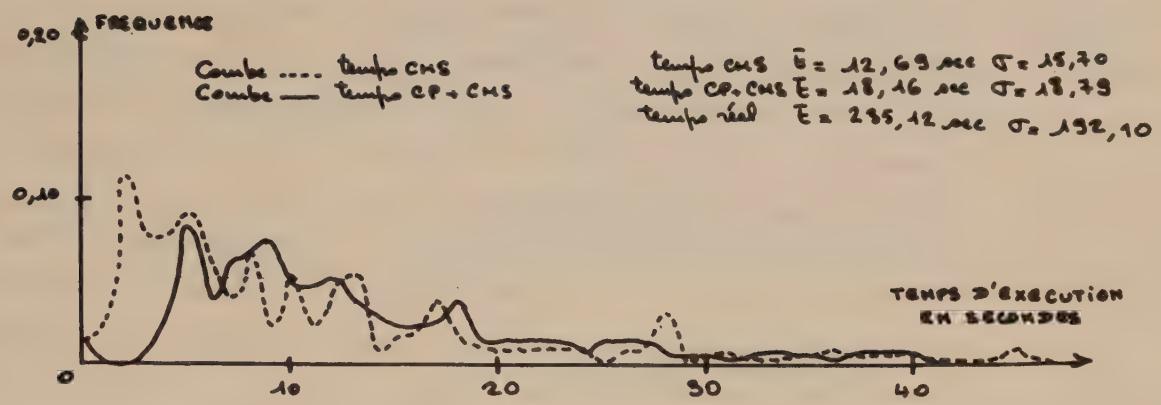
FIGURE 9

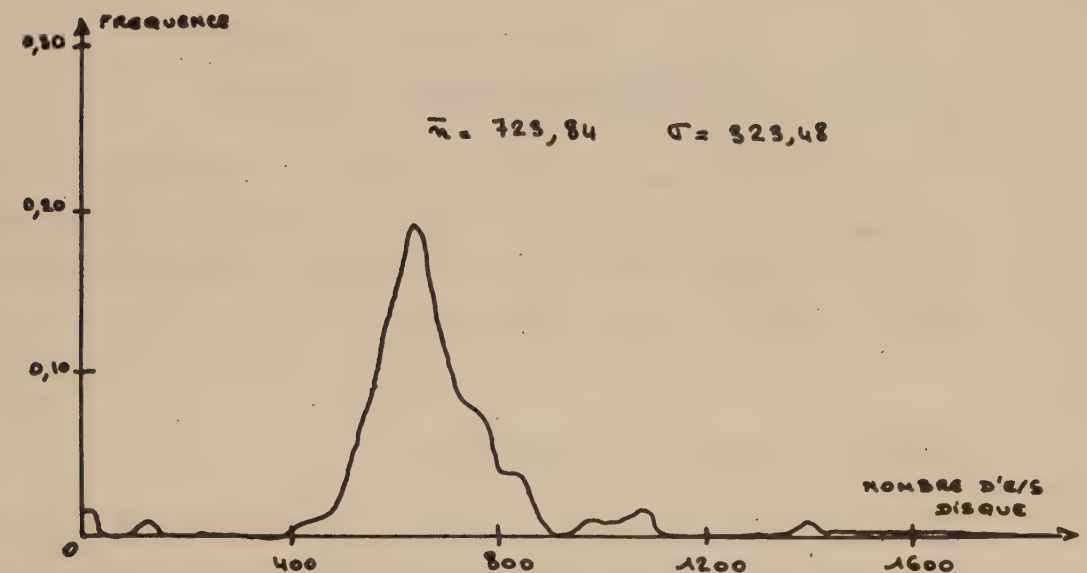
FIGURE 10


COMMANDES FORTRAN
(FORTRAN IV G IBM 360)
TAILLE DE L'ECHANTILLON 969
FREQUENCE
$\bar{n}$ = 2,72
σ = 4,96
NOMBRE D'E/S CONSOLE
$\bar{n}$ = 56,46
σ = 98,65
NOMBRE D'E/S DISQUE
TEMPS D'EXECUTION EN SECONDES
Courbe ---- temps CMS
Courbe —— temps CP+CMS
temps CMS $\bar{t}$ = 5,202 sec
σ = 7,558
temps CP+CMS $\bar{t}$ = 6,230 sec
σ = 8,295
temps réel $\bar{t}$ = 54,84 sec
σ = 65,82

COMMANDES COMPILATIONS PL/1

TAILLE DE L'ECHANTILLON 667





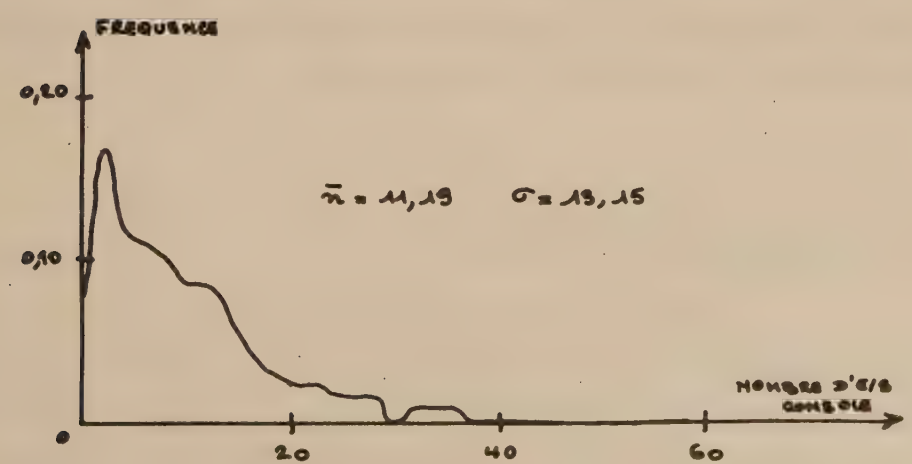


FIGURE 11

Applications de ces premiers résultats

a- *Améliorations du système CP à Grenoble*

L'équipe système de Grenoble a surtout été intéressée par la fréquence des différentes commandes CMS. Dans une version expérimentale, les tables du système ont été réorganisées en fonction de ces fréquences et les modules non résidents les plus utilisés ont été mis sur des supports plus rapides (tambour au lieu de disque)[39]. Ces modifications ont rendu les accès aux modules du système CMS 4 fois plus rapides et ont diminué de 5% le temps de WAIT (Unité Centrale en attente) au profit du temps problème (unité centrale fonctionnant en mode esclave).

b- *Algorithmes de scheduling "adaptés"*

Dans le système expérimental GMS [4], J.BELLINO et L.FINET ont implémenté un dispatcher ayant une hiérarchie de files d'attente. Suivant la durée probable de ses requêtes un utilisateur est mis dans la file d'attente optimale pour lui [15].

c- *Modèles de demandes et simulations*.

De tels résultats ont permis à C.BOKSENBAUM [5] de définir des modèles d'utilisation qui lui servent d'entrée dans un simulateur du système CP-67 et d'en déduire les réactions internes du système à une variation de l'utilisation.

3.2.3. Résultats pour OS/360 - MVT

Dans le système OS/360-MVT "batch", la notion analogue à une "commande" est le "step" du langage de contrôle (Job Control Language) [8]. Nous avons profité de l'outil des mesures SMF (System Management Facilities [21,22]) que fournit IBM avec les dernières versions d'OS pour procéder à une analyse analogue à celle faite dans

CP/CMS (Voir ANNEXE 2).

OS/360-MVT tourne sur la même configuration que CP/CMS (cf: ANNEXE 3), mais impose à l'unité centrale IBM 360 de fonctionner en mode 65 (pas de traduction dynamique des adresses -adressage classique-).

Nous donnons en ANNEXE 5 les résultats complets de l'analyse effectuée. Nous en extrayions les caractéristiques suivantes :

Période analysée	9-03 au 8-04-72
Nombre de "steps" exécutés	41 907
Temps moyen d'exécution d'un "step"	14,63 sec

Processeur	Fréquence	% Temps UC	Temps UC moyen sec	Ecart type
ALGOL 60	1,21 %	0,24 %	2,89	1,79
ALGOL W	2,03 %	2,22 %	16,01	38,79
ASSEMBLEUR	7,68 %	4,66 %	8,87	7,10
FORTRAN	7,77 %	3,48 %	6,54	9,89
COBOL	3,93 %	2,09 %	7,77	10,56
FORMAC	1,21 %	0,13 %	1,41	2,15
GPSS	0,06 %	0,14 %	34,36	27,92
PL/1	11,37 %	8,42 %	10,82	10,98
MPS	0,09 %	0,01 %	1,67	0,43
PL360	0,41 %	0,04 %	1,59	2,37
EDITEUR DE LIEN	22,48 %	4,75 %	3,08	114,77
EXECUTIONS UTILISATEUR	24,10 %	55,70 %	34,99	187,15

Pour avoir une vue plus globale de l'utilisation et de la charge du système, nous donnons le tableau suivant :

Signification	Compilations et Assemblages	Chargements et exécutions	Utilitaires
Fréquence	35,7 %	46,58 %	17,72 %
% Temps d'unité centrale	21,40 %	60,45 %	18,15 %

On remarque que la fréquence des compilations et assemblages est élevée, ce qui s'explique par le passage des Travaux Pratiques d'étudiants sous OS-MVT. On peut d'ailleurs estimer le pourcentage des compilations qui ne sont pas suivies d'une exécution :

$$35,70 - (22,48 + 2,90) = 10,32\ \%$$

(cf - ANNEXE 5)

3.3. Comparaisons de CP/CMS, OS/360-MVT et CTSS

Le niveau symbolique des mesures prélevées permet de comparer des systèmes très différents tant du point de vue de la finalité du système que de celui du matériel utilisé.

3.3.1. CP/CMS et OS/360-MVT

On remarque que dans OS on a :

- des compilations plus nombreuses, mais qui chargent moins l'unité centrale (bien qu'elles aient des temps d'exécution comparables)
- des exécutions de programmes utilisateurs moins nombreuses, mais qui chargent plus l'unité centrale.(Les exécutions sont de 2 fois plus longues que dans CP/CMS).

Ceci semble vérifier les hypothèses suivantes :

- beaucoup de compilations courtes sous OS dûes aux étudiants
- quelques compilations avec petites exécutions d'essai sous CP/CMS
- les utilisateurs exécutent de gros programmes sous OS après les avoir éventuellement mis au point d'une manière intéractive sous CP/CMS.

A ce sujet, on peut regretter ne de pas avoir pû disposer de la taille des fichiers "sources" sous OS. Une telle variable n'était pas disponible du fait qu'OS tourne sous ASP dans la configuration mesurée.

3.3.2. CP/CMS et CTSS

Le système CTSS est un système conversationnel en temps partagé qui tourne sur un IBM 7094 [5].

Alan Lee SCHERR a fait une analyse de ce système pour alimenter un simulateur. Nous tirons de sa thèse le tableau qui suit.

Le système analysé comportait 250 utilisateurs :

- période d'analyse: 29-12-64 au 4-02-65
- nombre de commandes prises en compte 80000

Signification	Manipulations de fichiers	Editions de fichiers	Exécutions Mises au point	Assemblages et compilations	Divers
Fréquence	35 %	15 %	12 %	9 %	29 %
% unité centrale	21 %	15 %	25 %	16 %	23 %
% temps réel	22 %	27 %	18 %	6 %	27 %

En rapprochant ce tableau de la figure 4, on remarque que si CTSS est comparable à CP/CMS du point de vue des fréquences des commandes, il n'en est pas de même pour la charge de l'unité centrale. Peut-être CTSS ne disposait-il pas d'un "éditeur" aussi puissant que celui de CP/CMS, d'où de nombreuses manipulations de fichiers, mais le temps passé par l'utilisateur devant sa console semble identique (22 + 27 = 49 %, contre 5 + 52 = 57 % pour CP/CMS).

CHAPITRE 4

<><*****><>

4. CORRELATIONS ET REGRESSIONS

4.1. Généralités

Notion "d'ensemble instantané de ressources nécessaires"

Après une analyse qui nous a permis de définir les caractéristiques de l'utilisation du système CP/CMS et de nous rendre compte, en particulier, que 50 % de la charge de l'unité centrale est constituée de processeurs standard, nous avons essayé d'étudier le comportement de ces processeurs.

Cette démarche a pour but la détermination du jeu de ressources nécessaires à un processeur ou à la requête d'un utilisateur. Les premières tentatives en ce sens ont porté sur l'utilisation de la mémoire paginée dans les systèmes en temps partagé, où on essaie de déterminer à l'instant t et durant l'intervalle de temps τ l'ensemble des pages nécessaires à un programme, appelé "working set" et noté $W(t,\tau)$ [10]. Pour nous, il s'agit de définir un "working set" généralisé que nous appellerons: "ensemble instantané de ressources nécessaires" et qui tient compte d'autres ressources que les pages en mémoire.

L'originalité ne tient pas tant aux variables considérées qu'aux relations qui sont créées entre elles. Tout système connait les "processeurs" qu'il gère et les paramètres qui y sont associés

(taille des fichiers manipulés, etc...), mais ces variables sont rarement reliées entre elles et jamais au nombre et type de ressources qu'elles mettent en jeu (temps d'unité centrale, entrées-sorties, etc...)

Nous nous sommes intéressés à deux problèmes particuliers :

- définir l'ensemble instantané des ressources nécessaires des assembleurs et compilateurs à partir de la taille du source
- définir le temps de réponse à une commande d'un utilisateur.

4.2. Corrélations

4.2.1. Rappels

On définit le coefficient de corrélation entre deux variables aléatoires X et Y par :

$$R(X,Y) = \frac{E((X - E(X))(Y-E(Y)))}{\sigma(X)\ \sigma(Y)}$$

où E(X) est l'espérance mathématique de la variable X et $\sigma(X)$ l'écart type de X (cf § 3.2.1.)

On démontre que :

- $|R(X,Y)| \leq 1$
- si X et Y sont indépendantes au sens des probabilités alors $R(X,Y) = 0$
- si $|R(X,Y)| = 1$ alors X et Y sont liées linéairement et réciproquement.

Si les variables X et Y prennent les valeurs x_i et y_i $i \in \{1,2,\ldots,n\}$, R(X,Y) est estimé par

$$r(X,Y) = \frac{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{s(X)\ s(Y)}$$

soit :

$$\frac{\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{X})^2} \times \sqrt{\sum_{i=1}^{n} (y_i - \bar{Y})^2}}$$

Ce qui peut encore s'écrire en développant le numérateur, en multipliant haut et bas par n et en remarquant que :

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i :$$

$$r(X,Y) = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \times \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \times \sqrt{n \sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}}$$

Cette dernière formule est très intéressante pour les calculs de r(X,Y), car elle permet de les effectuer par <u>un seul balayage des valeurs</u> x_i et y_i.

Il faut noter que ce problème de balayage des valeurs x_i et y_i est très important pour nous du fait des échantillons traités où n est grand.

4.2.2. Interprétation géométrique [37]

Le diagramme de dispersion des lois X et Y est formé des points (x_i, y_i) dans un système d'axes rectangulaires.

Lorsque tous les points ont tendance à se rapprocher d'une même droite, la corrélation est linéaire. Dans cette hypothèse, une relation linéaire sera toute indiquée pour exprimer l'estimation. Quand tous les points ont tendance à se rapprocher d'une même courbe, la corrélation est non linéaire et il faut alors rechercher une relation non linéaire.

Si Y varie dans le même sens que X, la corrélation est positive, si X et Y varient en sens inverse, la corrélation est négative.

Dans le cas d'une corrélation linéaire, on peut chercher à ajuster le nuage de points du diagramme de dispersion par une droite des moindres carrés.

C'est-à-dire qu'on recherche la droite

$$Y = a_1 X + b_1 \qquad (1)$$

telle que :

$$\sum_{i=1}^{n} (y_i - a_1 x_i - b_1)^2 \qquad \text{soit minimum}$$

ce qui revient à minimiser des distances sur l'axe des Y (écart entre l'ordonnée du point et l'ordonnée du point de la droite ayant même abscisse).

On dit alors que Y est la variable expliquée et X la variable explicative.

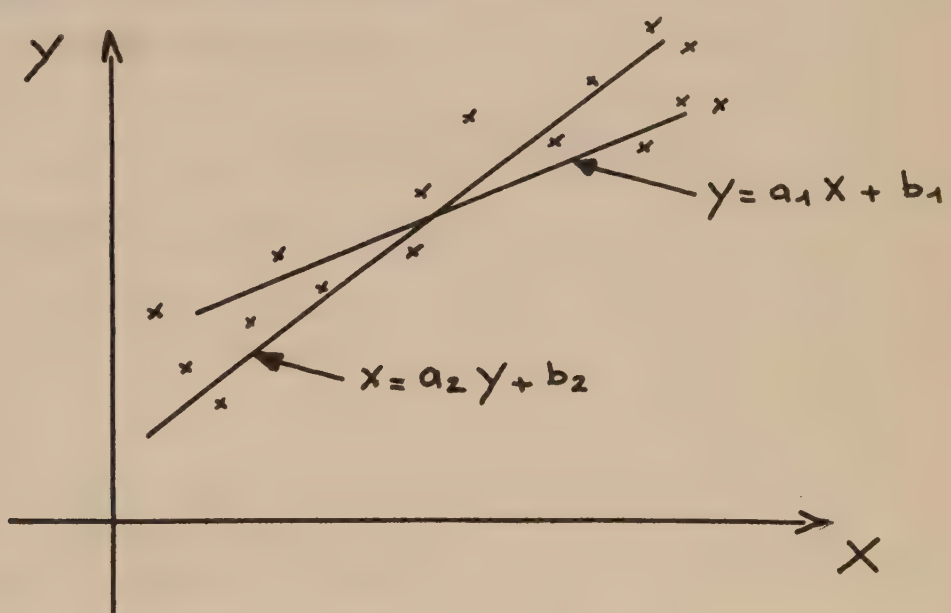
Si on permute X et Y, que X devient la variable expliquée et Y la variable explicative et que l'on recherche la droite des moindres carrés

$$X = a_2 Y + b_2$$

telle que

$$\sum_{i=1}^{n} (x_i - a_2 y - b_2)^2 \quad \text{soit minimum}$$

On obtient généralement une autre droite que (1), car on minimise alors des distances sur l'axe des X (écart entre l'abscisse du point et l'abscisse du point de la droite ayant même ordonnée)



On démontre alors [37] :

- que les pentes des 2 droites sont égales, si et seulement si $r(X,Y) = \pm 1$ (les 2 droites sont alors identiques)

- que si $r(X,Y) = 0$ les droites sont alors perpendiculaires et il n'y a pas de corrélation linéaire entre X et Y

- que $r^2(X,Y) = a_1 a_2$

Ainsi le coefficient de corrélation mesure l'écart entre 2 droites.

4.2.3. Applications et résultats

Nous avons calculé les coefficients de corrélation entre les variables mesurées pour déterminer :

- les relations existant entre ces variables
- la forme de ces relations

Nous nous sommes intéressés :

- aux assembleurs et compilateurs
  - . ASSEMBLEUR G IBM 360
  - . FORTRAN IV G
  - . PL/1 F
- à l'ensemble de toutes les commandes
- et à l'éditeur qui est le composant le plus utilisé (cf.ANNEXES 4 et 6)

Editeur

	Temps réel	Longueur fichier	E/S console	E/S disque	Temps virtuel	Overhead
Temps réel	1	0,14	0,84	0,41	0,49	0,71
Longueur fichier	0,14	1	0,18	0,38	0,41	0,29
E/S console	0,84	0,18	1	0,38	0,47	0,74
E/S disque	0,41	0,38	0,38	1	0,85	0,80
Temps virtuel	0,49	0,41	0,47	0,85	1	0,88
Overhead	0,71	0,29	0,74	0,80	0,88	1

REMARQUES

- La longueur du fichier édité est indépendante des autres variables
- L'overhead provient essentiellement de la simulation des E/S par CP (E/S disque et console).
- Le temps de réponse est déterminé par les E/S console.

Fortran IV G

	Temps réel	Longueur fichier	E/S console	E/S disque	Temps virtuel	Nb utilisateurs	Nb util* Temps CP
Temps réel	1	0,51	0,64	0,57	0,58	0,27	0,65
Longueur fichier	0,51	1	0,59	0,82	0,90	-	-
E/S console	0,64	0,59	1	0,61	0,63	-	-
E/S disque	0,57	0,82	0,61	1	0,86	-	-
Temps virtuel	0,58	0,90	0,63	0,86	1	-	-
Nbre utilisateurs	0,27	-	-	-	-	1	-
Nbre util* temps CP	0,65	-	-	-	-	-	-

REMARQUES

- La longueur du fichier compilé doit permettre de déterminer le temps virtuel de compilation, le nombre d'E/S disque est dans une moindre mesure le nombre d'E/S console (ce qui est normal, car les E/S consoles proviennent de diagnostics d'erreur du compilateur)
- Le temps de réponse (temps réel) dépend plus de Nb util*Temps CP que Nombre utilisateurs seul (apparition du facteur d'élongation).

PL/1

	Temps réel	Longueur fichier	E/S console	E/S disque	Temps virtuel	Nombre utilisateurs	Nbre util* Temps CP
Temps réel	1	0,43	0,55	0,49	0,56	0,31	0,71
Longueur fichier	0,43	1	0,65	0,54	0,86	-	-
E/S console	0,55	0,65	1	0,64	0,79	-	-
E/S disque	0,49	0,54	0,54	1	0,70	-	-
Temps virtuel	0,56	0,86	0,79	0,70	1	-	-
Nbre utilisateurs	0,31	-	-	-	-	1	-
Nbre util* Temps CP	0,71	-	-	-	-	-	1

REMARQUES

- La longueur du source permet de déterminer le temps de compilation et, paradoxalement, le nombre d'E/S console mieux que le nombre d'E/S disque (le compilateur PL/1 fait beaucoup d'E/S disque pour lui-même de par sa structure d'overlay).
- Présence pour le temps réel d'un facteur d'élongation.

Assembleur 360 G

	Temps réel	Longueur fichier	E/S console	E/S disque	Temps virtuel	Nb utili-sateurs	Nb util* Temps CP
Temps réel	1	0,53	0,56	0,76	0,82	0,31	0,88
Longueur fichier	0,53	1	0,37	0,64	0,74	-	-
E/S console	0,56	0,37	1	0,42	0,47	-	-
E/S disque	0,76	0,64	0,42	1	0,91	-	-
Temps virtuel	0,82	0,74	0,47	0,91	1	-	-
Nb utili-sateurs	0,31	-	-	-	-	1	-
Nb util* temps CP	0,88	-	-	-	-	-	1

Toutes les commandes

	Temps réel	E/S console	E/S disque	Temps virtuel	Overhead	Nb utili-sateurs	Nb util* Temps CP	Temps CP
Temps réel	1	0,80	0,31	0,38	0,52	0,02	0,45	0,42
E/S console	0,80	1	0,17	0,34	0,39	-	-	0,65
E/S disque	0,31	0,17	1	0,25	0,63	-	-	0,35
Temps virtuel	0,38	0,34	0,25	1	0,50	-	-	0,98
Overhead	0,52	0,39	0,63	0,50	1			0,64
Nb utilisa-teurs	0,02	-	-	-		1		
Nb utilis. Temps CP	0,45	-	-	-			1	
Temps CP	0,42	0,65	0,35	0,98	0,64	-	-	1

REMARQUES

Les liaisons sont moins importantes du fait du mélange de toutes les commandes, mais on note que :

. le temps de réponse dépend surtout des E/S console
. le facteur d'élongation est peu sensible sans doute du fait du grand nombre de commandes courtes
. l'overhead provient surtout des E/S disque.

4.2.4. Conclusions

Il faut être très prudent dans l'interprétation de ces résultats où un coefficient de corrélation élevé

- peut être dénué de sens (corrélation trompeuse)
- ou indiquer une liaison au second niveau [7]
  . la liaison constatée entre X et Y provient de liaisons de X et Y avec une troisième variable Z mesurée ou non (variable latente ou cachée).

Dans notre cas, nous pouvons quand même conclure :

a- Compilateurs et assembleurs :

- les ressources employées sont liées à la taille de "Source"
- les E/S console, qui viennent des diagnostics d'erreurs, montrent que plus un programme est long plus il y a d'erreurs!

b- Facteur d'élongation: Temps UC × Nb utilisateur

Ce facteur est une bonne indication de la manière dont s'effectue le partage de temps entre les utilisateurs. Avec un partage classique par tranches de temps, si chaque utilisateur utilise pleinement sa

tranche, chacun a l'impression que son temps d'exécution a duré

Temps UC × Nb utilisateur

On remarque que, dans CP/CMS, ce partage de temps est effectif pour les commandes longues (compilations, etc..) mais ne l'est pas pour les commandes courtes et toutes les commandes en général.
Il faut en déduire que le partage de temps s'effectue essentiellement par d'autres mécanismes que l'attribution de tranches de temps. Il se fait surtout par l'exploitation de l'asynchronisme de l'unité centrale et des entrées-sorties dûes :

- aux E/S lancées par les programmes
- à la pagination

4.3. Régressions [11,12]

4.3.1. Le modèle

On dispose de n+1 variables aléatoires

$$Y, X_1, X_2, \ldots, X_n$$

On veut'expliquer' Y par $X_1, X_2, \ldots, X_n$ avec une relation de la forme :

$$Y = \sum_{i=1}^{n} a_i X_i$$

pour introduire une constante a_o dans cette relation, nous considérons aussi la variable aléatoire X_o, qui vaut toujours 1, on a alors :

$$Y = \sum_{i=o}^{n} a_i X_i$$

Supposons que X_o, X_1,..., X_n ,Y appartiennent à l'espace vectoriel des fonctions de carré sommable. Cet espace est euclidien muni du produit scalaire :

$$< X,Y > = E(XY)$$

qui définit la norme

$$||X|| = \sqrt{< X,X >} = \sqrt{E(XX)} = \sqrt{E(X^2)}$$

On peut écrire :

$$Y = \sum_{i=o}^{n} a_i X_i + \varepsilon$$

avec ε appartenant à l'espace des fonctions de carré sommable et $\forall i \in \{1,...,n\}$, $a_i \in R$

On va rechercher les a_i tels que le terme d'erreur ε soit minimum :

$$||\varepsilon|| \text{ minimum}$$

On recherche donc les a_i tels que

$$||Y - \sum_{i=o}^{n} a_i X_i|| \text{ soit minimum}$$

Pour cela il faut et il suffit que (cf [26]):

$$\forall j \in \{o,...n\} \quad \frac{\partial}{\partial aj} ||Y - \sum_{i=o}^{n} a_i X_i|| = 0$$

La dérivation est au sens des normes sur R et on a une norme sur un espace de variables aléatoires. Il faut donc développer pour pouvoir dériver

$$\forall j \in \{o,...,n\} \quad \frac{\partial}{\partial aj} E((Y - \sum_{i=o}^{n} a_i X_i)^2) = 0$$

L'opérateur E (intégrale au sens de Lebesgue) a les propriétés voulues pour écrire :

$$\forall j \in \{o,\ldots,n\} \quad E\left(\frac{\partial}{\partial aj}\left(Y - \sum_{i=o}^{n} a_i X_i\right)^2\right) = 0$$

(dérivation sous le signe somme)

$$\forall j \in \{o,\ldots,n\} \; E\left(-2\left(Y - \sum_{i=o}^{n} a_i X_i\right) X_j\right) = 0$$

E est linéaire

$$\forall j \in \{o,\ldots,n\} \; \sum_{i=o}^{n} a_i E(X_i X_j) = E(YXj)$$

On obtient un système de n+1 équations avec un déterminant de GRAM (non nul si X_i indépendants [26])

Si on se souvient des propriétés de X_o :

$$E(X_o X_i) = E(X_i) \quad \text{et} \quad E(X_o^2) = 1$$

le système devient :

$$a_o + E(X_1)a_1 + \ldots + E(X_n)\, a_n = E(Y)$$

$$E(X_1)a_o + E(X_1^2)a_1 + \ldots + E(X_1 X_n)a_n = E(X_1 Y)$$

$$\vdots$$

$$E(X_n)a_o + E(X_n X_1) + \ldots + E(X_n^2)\, a_n = E(X_n Y)$$

$E(X_i X_j)$ est la covariance des variables X_i et X_j. La matrice de résolution du système est donc la matrice de covariances des variables explicatives. Il faut remarquer que si les variables sont centrées (E(X) = o) et réduites ($\sigma(X) = 1$) ce qui est toujours possible par le changement

$X' = \dfrac{X - E(X)}{\sigma(X)}$, on a alors :

$$E(XY) = R(X,Y)$$

La matrice de covariances est aussi la matrice de corrélations.

Notons aussi que :

- si les lois sont centrées: $E(X_i) = E(Y) = 0$
  alors $a_o = 0$ (première équation du système)
- $E(\varepsilon) = 0$ (ε est centrée). En effet :

$$E(\varepsilon) = E(Y - \sum_{i=o}^{n} a_i X_i) = E(Y) - \sum_{i=o}^{n} a_i E(X_i) = 0$$

(première équation du système).

De telles remarques nous ont permis de contrôle l'exactitude des résultats trouvés .

Interprétation statistique et géométrique

Si les variables $Y, X_1, \ldots X_n$ sont estimées par les valeurs $\{y_1, \ldots y_m\}$, $\{x_{1,1} \cdots x_{1,m}\}, \ldots \{x_{n,1}, \ldots x_{n,m}\}$, la matrice de résolution du système est estimée par :

$$a_o + a_1 \sum_{i=1}^{m} x_{1,i} + \ldots + a_n \sum_{i=1}^{m} x_{n,i} = \sum_{i=1}^{m} y_i$$

$$a_o \sum_{i=1}^{m} x_{1,i} + a_1 \sum_{i=1}^{m} x_{1,i}^2 + \ldots + a_n \sum_{i=1}^{m} x_{n,i}\, x_{1,i} = \sum_{i=1}^{m} x_{1,i}\, y_i$$

$$\vdots$$

$$a_o \sum_{i=1}^{m} x_{n,i} + a_1 \sum_{i=1}^{m} x_{n,i}\, x_{1,i} + \ldots + a_n \sum_{i=1}^{m} x_{n,i}^2 = \sum_{i=1}^{m} x_{n,i}\, y_i$$

C'est ce système que nous avons résolu pour trouver les coefficients de nos relations.

On peut remarquer que ce système est la généralisation de celui qui permet de déterminer les coefficients de la droite "des moindres carrés" dans un espace à 2 dimensions.

On recherche ici, dans un espace n + 2 dimensions, l'hyperplan qui ajuste le mieux le nuage de points.

4.3.2. Les buts et les critères associés

a- But explicatif

Les coefficients a_i calculés indiquent l'importance de chaque variable X_i (ou expression de variables) sur la variable Y. Une telle interprétation des coefficients a_i est valable [7] :

- si les variables X_i sont indépendantes (dans notre modèle, l'indépendance linéaire découle de l'indépendance au sens des probabilités)
- s'il n'existe pas de variables importantes non mesurées influençant les variables mesurées (influence de ε sur les X_i)

ULMO [35] donne des méthodes pour choisir les variables explicatives par rapport aux variables à expliquer. Nous nous contenterons de choisir des variables explicatives peu corrélées entre elles et très corrélées à la variable pour expliquer.

b- But prédictif

La relation $Y = f(X_1, \ldots, X_n)$ permet, connaissant les valeurs de $X_1, \ldots, X_n$, de prédire avec une bonne approximation la valeur de Y.

Suivant les coefficients de corrélation, les variables considérées pourront être des expressions de variables mesurées (par exemple : X_i X_j). L'équation trouvée ne sera plus celle d'un hyperplan, mais d'une surface gauche dans un espace à n+2 dimensions (n étant le nombre de variables considérées).

c- Les critères associés

Comme critères de validité des relations, nous considèrerons :

- un critère explicatif: le coefficient de corrélation multiple entre Y et $X_1 \ldots X_n$: R
avec

$$R^2 = \frac{\text{Variance expliquée}}{\text{Variance totale}}$$

$0 \leq R \leq 1$. Plus R est proche de 1, plus les variables $X_1, \ldots, X_n$ "expliquent" correctement Y

- un critère prédictif : l'écart moyen normé sur Y.
Si Y est la valeur réelle

$$Y = f(X_1, \ldots, X_n) + \varepsilon$$

L'écart moyen normé sera : $\frac{\overline{|\varepsilon|}}{\overline{Y}}$

4.3.3. Les résultats et interprétations

Dans ce qui suit, parmi les nombreuses relations qui ont été calculées, nous donnons celles qui ont semblé les meilleures.

TR	désigne le temps réel en secondes
TCMS	désigne le temps virtuel en secondes
TCP	désigne le temps virtuel + temps CP en secondes
ESDIS	désigne le nombre d'E/S disque
ESCON	désigne le nombre d'E/S console
NB	désigne le nombre d'utilisateurs simultanés
OVH	désigne l'overhead = TCP - TV
SOURCE	désigne le nombre de cartes du fichier "source".

COMMANDES CONCERNEES	RELATIONS CONCERNEES	Coefficient de corrélation multiple	Ecart moyen normé
EDIT	OVH = 0,10+0,56xTCMS+0,003xESDIS+0,012xESCON	0,92	16,7 %
FORTRAN IV G	TCMS = -0,32 + 0,036 x SOURCE	0,93	30 %
	ESCON = 0,34 + 0,015 x SOURCE	0,71	60 %
	ESDIS = 30,81 + 0,165 x SOURCE	0,94	15,8 %
	TCMS = -2,30+0,023xSOURCE+0,14xESCON+0,062xESDIS	0,95	28,4 %
PL/1 F	TCMS = 4,82 + 0,03 x SOURCE	0,92	28 %
	ESCON = 6,02 + 0,019 x SOURCE	0,82	52,5 %
	ESDIS = 616,69 + 0,418 x SOURCE	0,94	15,4 %
	TCMS = -3,20+0,02xSOURCE+0,33xESCON+0,01xESDIS	0,96	24,4 %
ASSEMBLEUR 360 G	TCMS = 3,60 + 0,014 x SOURCE	0,87	42 %
	ESCON = 3,66 + 0,01 x SOURCE	0,58	71,6 %
	ESDIS = 538,63 + 1,346 x SOURCE	0,83	48,5 %
	TCMS = -0,11+0,0049xSOURCE+0,054xESCON+0,0065xESDIS	0,96	16 %
TOUTES LES COMMANDES	TR = 7,62+0,12xESDIS+5,75xESCON-4,77xTCP+0,21xTCPxNB	0,85	54 %
	OVH = 0,085+0,070xTCMS+0,0058xESDIS+0,019xESCON	0,64	43,5 %

Une relation très intéressante à commenter est celle que nous donnons pour le temps de réponse :

$$TR = 7,62 + 0,12 \times ESDIS + 5,75 \times ESCON - 4,77 \times TCP + 0,21 \times TCP \times NB$$

Nous en déduisons :

- qu'une E/S disque prend en moyenne
  0,12 sec
- et qu'une E/S console dure en moyenne
  5,75 sec

Ce qui parait correspondre à la réalité.

Il est même possible d'interpréter le terme :

$$- 4,77 \times TCP + 0,21 \times TCP \times NB$$

en mettant 0,21 x TCP en facteur :

$$0,21 \times TCP \ (NB - 22,7)$$

Si on se souvient que 34,4 utilisateurs sont en moyenne sur le système, on en déduit qu'il n'y en a que :

$$34,4 - 22,7 = 11,7$$

qui soient actifs et participent effectivement au partage de temps, soit environ 1/3 des utilisateurs.

On peut penser que les 2/3 des utilisateurs, qui sont inactifs, sont en attente d'E/S console, car les 2/3 du temps réel d'exécution d'une commande est passé en moyenne en E/S console.

En effet, une commande dure en moyenne

69 sec (figure 5)

et comporte 8,17 E/S console qui prennent

5,75 sec chacune (fig 5 et ci-dessus)

le temps pris par les E/S console est donc de :

$$8,17 \times 5,75 = 46,98 \text{ sec}$$

ainsi

$$\frac{46,98}{69} \simeq \frac{2}{3} \text{ : les } \frac{2}{3} \text{ du temps}$$

l'utilisateur est en attente d'E/S console.

De tels résultats ne semblent pas aberrants et permettent de comprendre pourquoi le degré en multiprogrammation est si bas : 2,5 à 4 (mesures directes faites par RODRIGUEZ [30]).

4.3.4. D'autres méthodes d'analyse

On remarque que la détermination de l'ensemble instantané des ressources nécessaires d'une commande manque de précision. Les relations trouvées sont d'autant moins bonnes qu'elles portent sur des groupes de commandes plus diversifiées. Même les regroupements "logiques" semblent peu satisfaisants.

A partir de cette constatation, deux démarches sont possibles :

Création de variables formelles

On crèe des variables formelles mieux adaptées à l'analyse des données. Pour ce faire, on peut se servir des propriétés des valeurs propres d'une matrice de covariances, afin de déterminer les variables formelles essentielles, dites "composantes principales" [24] : ce sont les vecteurs propres associés aux valeurs propres.

Cette méthode permet :

- d'avoir des variables explicatives indépendantes
- de réduire le nombre des variables explicatives en éliminant celles qui sont associées à des valeurs propres très petites (théorème de GUTTMAN)

et peut fournir des explications satisfaisantes :

- par exemple, expliquer un temps de compilation par les 2 variables formelles X'_1 et X'_2 interprétées comme "la taille de source" et "les erreurs de compilations".

Cette méthode n'est pourtant pas exploitable par nous directement: elle ne peut avoir qu'un rôle directif.

Création de "types" formels

On crèe des "types" formels de commandes, d'utilisateurs et de ressources. Au lieu de regrouper "logiquement" des commandes ou des utilisateurs, on définit une distance permettant de constituer des nuages de points qui forment des "types".

Les difficultés majeures de cette méthode sont :

- définir une métrique "intéressante"
- trouver des "bons"algorithmes de constitution de nuages
- interpréter les "types" trouvés (par exemple, il se peut qu'une commande courte de compilation soit du même "type" qu'une commande longue de manipulations de fichiers).

Applications de la typologie

Reprenons notre modèle de système en couches (cf § 1.1). Pour essayer de prévoir et résoudre, au niveau de la couche 4, les problèmes qui se posent dans un système (pagination, E/S, interblocages, etc...), nous envisageons la démarche suivante :

- on considère qu'un élément de la couche 4 (processeur) est constitué d'éléments de la couche 3 (ressources logiques)

- on fait une typologie au niveau de la couche 4.

A partir de là on met en place des algorithmes de scheduling :

- qui tiennent compte de la typologie trouvée en gérant les divers "types"

- sont prédictifs
  . au niveau des "types" en associant à chacun le jeu de ressources qu'il emploie
  . à l'intérieur d'un "type" par l'intermédiaire de relations analogues à celles que nous avons déterminées (elles seraient beaucoup plus précises du fait de l'homogénéité d'un "type").

Notons qu'on peut faire une typologie au niveau de la couche 5 (utilisateurs) en se servant de celle faite dans la couche 4 (double typologie).

Nous projetons de mettre en oeuvre "La méthode des nuées dynamiques" de E.DIDAY [11] sur nos données pour effectuer une telle analyse.

C O N C L U S I O N

L'analyse des mesures que nous venons de décrire débouche sur plusieurs applications concrètes :

- améliorations du système étudié par la connaissance qu'on a des composants les plus employés

- créations de modèles d'utilisation

  . pour créer des benchmarks qui reconstituent jusque dans les composants employés ce que fait un utilisateur

  . pour alimenter des simulateurs de systèmes.

- aide à la conception de nouveaux systèmes

  . systèmes spécialisés aux services éventuellement limités (un système d'édition, quand on sait que 50 % du temps de l'utilisateur est passé en édition)

  . systèmes auto-adaptatifs, l'adaptation pouvant être soit statique (à la génération du système après analyse de son environnement), soit dynamique si le système prend des mesures qui inter-agissent sur lui.

L'auto-adaptation est déjà réalisée de manière embryonnaire dans de nombreux systèmes (CP-67, MTS, OS/360 MVT [29], TENEX sur PDP10, etc...) où elle s'effectue essentiellement à un niveau interne (couches 2 et 3).

Des variables internes au système (fautes de pages, E/S, etc...) sont accumulées dans des compteurs et exploitées périodiquement par les algorithmes de scheduling. Ces algorithmes sont hiérarchisés :

- par ordre de complexité croissante
- par fréquence d'activation décroissante

Exemple : le niveau 1 est utilisé en permanence - simple algorithme FIFO
le niveau 2 n'est activé que toutes les 500 millisecondes -algorithmes plus complexes.
et le niveau 3 est activé toutes les secondes.

On peut bien sûr imaginer des critères d'activation fondés sur une autre variable que le temps (nombre d'occurrences d'évènements).

Nous proposons deux démarches distinctes suivant les systèmes concernés.

Pour les petits systèmes, les systèmes spécialisés et ceux dont l'environnement et l'utilisation sont stables, on peut concevoir des mécanismes de prises de mesures et d'auto-génération pour avoir une adaptation statique à la génération du système. Il faut noter que cette optique intéresse la majorité des ordinateurs exploités industriellement.

Pour les gros systèmes à utilisation diversifiée, on peut, par contre, envisager une véritable auto-adaptation. Ce n'est possible à un coût raisonnable qu'avec des outils de mesures intégrés tant au niveau hardware que software [31] .

Les voies ouvertes dans ce domaine sont de deux ordres.

Du point de vue de la technologie, on peut songer à disposer :

Dans la couche 1

- d'un BCU accessible à partir de l'unité centrale par des instructions privilégiées, éventuellement du type E/S
- d'instructions spéciales, du type MONITOR CALL de la série IBM 370 [23], permettant de "tracer" l'exécution du système et d'entretenir des compteurs software.

Dans la couche 2-

- d'un dispatcher gérant l'asynchronisme technologique (sauvegarde de contextes et algorithmes simples : FIFO, LIFO ou LRU) avec les compteurs hardware associés (âge des pages, pour LRU). Ces mécanismes sont cablés ou tout au moins microprogrammés (cf: expérience du T1600 de la Télémécanique).

Dans les couches 3 et 4

- d'un scheduler
  - . à niveaux dans un contexte monoprocesseur
    Ce scheduler peut en particulier recevoir le contrôle, lorsque l'unité centrale va être mise en attente (entre 20 % et 30 % du temps d'attente dans de nombreux systèmes [30,17]).
  - . dans une unité spécialisée dans un contexte multiprocesseur (cf: expériences faites sur CONTROL DATA 6600 [33,34] et aboutissant à un scheduling prédictif [33]

Du point de vue des algorithmes, on peut penser à utiliser la typologie :

- par des algorithmes d'analyse typologique exploitables en temps réel. Ces algorithmes tiennent leur efficacité des connaissances acquises par une analyse typologique approfondie effectuée en différé auparavant.

- par des algorithmes de gestion du système utilisant les "types" créés et les relations prédictives à l'intérieur de ceux-ci.

BIBLIOGRAPHIE

REFERENCES

<>******<>

[1] B.ARDEN and BOETTNER — "Measurement and performance of a multiprogramming system". Second symposium of Operating Systems Principles. 1969.

[2] A.AUROUX et C.HANS — "Introduction aux systèmes CP-67 et CMS" Monographies informatiques -AFCET- Les systèmes conversationnels - L.BOLLIET DUNOD.

[3] Y.BARD — "CP-67 Measurement and analysis: overhead and throughput". Workshop on system performance evaluation April 1971 - Harward.

[4] J.BELLINO et Ph.POTIN — "Mécanismes d'un hyperviseur" Congrès AFCET 1972

[5] C.BOKSENBAUM — "Problèmes de la simulation des systèmes" Séminaires de l'IMAG. Grenoble Mai 1971

[6] D.T.BORDSEN — "Univac 1108 Hardware instrumentation system". Workshop on system performance evaluation April 1971 - Harvard.

[7] G.E.P. BOX
"Du bon et du mauvais usage de la regression".
Revue de statistique appliquée 1966. Vol XIV n°3.

[8] G.D. BROWN
"System/360 - Job Control Language"
John Wiley and Son.

[9] P.J.DENNING
"Virtual Memory"
Computing Surveys. Vol 2, n°3 - 1970

[10] P.J.DENNING
"The working set model for program behavioz"
Comm.ACM Vol 11, n°5 (May 1968).

[11] E.DIDAY
"Une nouvelle méthode en classification automatique et reconnaissance des formes La méthode des nuées dynamiques".
Revue de Statistique Appliquée. 1971.
Vol.XIX n°2

[12] E.W.DIJKSTRA
"The structure of THE multiprogramming system".
Comm.ACM Vol 11 n°5 (May 1968)

[13] J.P.DUPUY
"Description, fonctionnement, utilisation du Basic Counter Unit"
Note interne IMAG - Grenoble

[14] M.EZEKIEL and K.A.FOX
"Methods of correlation and regression analysis linear and curvilinear"
John Wiley and Son (1965)

[15] L.FINET et J.LEROUDIER — "Analyse de la demande d'un système. Application au scheduling dans un système avec contraintes de swapping". Séminaires de l'IMAG. Grenoble, Janvier 1973.

[16] F.A.GRAYBILL — "An introduction to linear statistical models". Mc Graw-Hill Book Company(1961)

[17] G.M.HOLTWICK — "Designing a commercial performance measurement system" Workshop on system performance evaluation April 1971 - Harvard.

[18] IBM Corp — "CMS Program Logic Manual" Form GU20 - 0591

[19] IBM Corp — "CP-67 Program Logic Manual" Form GY20-0590.

[20] IBM Corp — "IBM System/360 (Model 67)- Functional characteristics". Form A27 - 2719

[21] IBM Corp — "IBM System/360 Operating System Programmer's guide". Form GC28 - 6550

[22] IBM Corp — "IBM System/360 Operating System System Management Facilities(SMF)" Form GC28 - 6712.

[23] IBM Corp — "IBM System/370 - Principles of operation" Form GA22 - 7000

[24] L.LEBART,J.P.FENELON — "Statistique et informatique appliquées" DUNOD - Paris 1971.

[25] J.LEROUDIER — "Analyse d'un système à partage de ressources". Séminaires de 1'IRIA - Décembre 1972 - PARIS.

[26] Y.V.LINNIK — "Méthode des moindres carrés". DUNOD - Paris.

[27] H.C. LUCAS — "Performance evaluation and monitoring". Computing Surveys. Vol 3. n°3. Sept.1971.

[28] M.H.Mac DOUGALL — "Computer system simulation: an introduction" Computing Surveys - Vol 2- n°3 Sept 1970

[29] B.S. MARSHALL — "Dynamic calculation of dispatching priorities under OS/360 MVT". Datamation - August 1969.

[30] J.RODRIGUEZ, J.P.DUPUY — "Instrumentation for the evaluation of a time-sharing page demand system" Centre scientifique IBM de Grenoble.

[31] J.H.SALTZER,J.W.GINTELL — "Instrumentation of Multics" Comm.ACM Vol 13 - n°8 - Aug 1970.

[32] A.L.SCHERR — "An analysis of time-shared Computer system".Thesis. Project MAC -M.I.T. (1965)

[33] S.SHERMAN, F.BASKETT III, J.C. BROWNE — "Trace-Driven Modeling and Analysis of CPU scheduling in a multiprogramming system". Comm ACM - Vol 15 - n°12 Dec 1972

[34] D.F. STEVENS — "On overcoming high-priority paralysis in multiprogramming systems: a case history". Comm ACM - Vol 11 - n°8 Aug 1968.

[35] J.ULMO — "Problèmes et programmes de régression" Revue de Statistique Appliquée 1971 Vol XIX - n°1

[36] R.W. WATSON — "Time-sharing system design concepts" Mc Graw-Hill Book Company.

[37] M.R.SPIEGEL — "Theory and problems of STATISTICS" Mc Graw-Hill - New York - 1972

[38] CONTROL DATA CORP — "SCOPE System programmer's reference manual". SCOPE 3.4.

[39] X.DE LAMBERTERIE — "Espace virtuel et gestion de fichiers". Thèse à paraître. Grenoble.

A N N E X E S

A N N E X E 1

<>¤¤¤¤<>

LES MESURES DANS CP/CMS

FORMAT DES MESURES RECUEILLIES

Les mesures recueillies ont un format de cartes et il y a deux formats possibles.

1. Format standard

```
    8        8      4    4      8        8      4   3     6      6
AAAAAAAA BBBBBBBB CCCC DDDD EEEEEEEE FFFFFFFF GGGG HHH I JJJJJJ KKKKKK
   5      6      2    3
LLLLL MMMMMM N O PP Q RRR
```

A nom de la machine virtuelle qui a émis la commande

B nom de la commande (en toutes lettres, même si elle était abrégée)

C nombre de fautes de pages

D nombre d'E/S disques

E nom du fichier concerné

F type du fichier concerné

G longueur du fichier concerné (ou taille mémoire du module s'il s'agit d'un fichier de type MODULE)

H nombre d'E/S console

I { 0 commande non abrégée
  1 commande abrégée

J temps virtuel (CMS) en 1/100° de seconde

K temps total(CMS + CP) en 1/100° de seconde

L temps réel (devant le terminal, entre le retour chariot et la fin de la commande : message R;... ou E();...) en secondes

M heure de fin de la commande en heures et minutes et secondes

N { 0 commande sans erreur
  1 commande erronée

O Niveau d'imbrication ou de récursion de la commande (fichiers EXEC)

P Nombre d'utilisateurs du système CP/CMS

Q = 1 fichier en création

R nombre de pages en mémoire (pour la commande)

Toutes les chaînes de caractères (noms de commandes ou de fichiers) sont cadrées à gauche et tous les nombres (nombre d'E/S, temps, heure, etc...) sont cadrés à droite . Ceci pour faciliter un traitement et une analyse des données par des programmes écrits en FORTRAN seul langage dont tous les interfaces avec des unités telles que traceur de courbe et écran cathodique, sont disponibles).

REMARQUE

> Comme il n'est pas possible d'inclure dans INIT (module résident du noyau CMS) un décodeur trop gros, toutes les commandes ne sont pas décodées. Dans ce cas, les champs E, F, G, peuvent rester vides, mais les autres champs sont toujours remplis.

Liste des commandes décodées

OFFLINE
CEDIT
EDIT
ERASE
PRINTF
ASSEMBLE
ASMG
ALGOL
PLI
FORTRAN
LOAD
LOADMOD
PL360NEW
SCRIPT
SNOBOL
UPDAT
$
COMBINE

ainsi que toutes les macro-commandes (implicites ou explicites) et les commandes utilisateur appelant un module utilisateur.

Parmi la liste des commandes précédentes, on ne décode pas celles qui se servent de paramètres multiples :

Ex : ERASE * * (effacer les fichiers de n'importe quel nom et n'importe quel type)
ASMG TOTO JULE (assembler les fichiers TOTO et JULE)

Lorsqu'une commande n'est pas décodée, son nom est mis dans B et les deux paramètres qu'elle peut avoir dans E et F (on sauve ainsi les meubles).

2. Format disque

Lorsqu'un utilisateur CMS frappe LOGIN (explicitement ou implicitement) une carte spéciale portant comme nom de commande DISQUE est alors émise.
Cette carte décrit la configuration des disques de la machine CMS (cette configuration ne peut varier que par des commandes LOGIN) et l'occupation de ces disques.

```
   8
ZZZZZZZZ DISQUE  /_ / /_/  P X YYYYY VVVVV T X YYYYY VVVVV A X ....
```

Z nom de la machine virtuelle
X indique si le disque dont l'indicatif précède X est en extension

Ex : X = blanc pas d'extension
X = P le disque est en extension de P

Y nombre d'enregistrements physiques (800 octets) utilisés
V nombre d'enregistrements " " libres

REMARQUE

Si après IPL CMS l'utilisateur frappe une commande autre que LOGIN, il y a alors LOGIN implicite (LOGIN 191 P) et la carte DISQUE ne sera émise qu'après la commande demandée.

A N N E X E 2

<>xxxxx<>

LES MESURES DANS OS/360-MVT

PRINCIPE DE SMF [21,22]

Les dernières versions d'OS/360-MVT possèdent un outil de mesures : SMF (System Management Facilities) qui délivre 31 sortes d'enregistrements.

Nous reprenons la terminologie IBM :

N° de code de l'enregistrement	Enregistrement
0	IPL
1	Wait time
2	Dump header
3	Dump trailer
4	Step termination
5	Job termination
6	Output writer - class SYSOUT
7	Data last
8	I/O configuration
9	VARY ONLINE
10	Allocation recovery
11	VARY OFFLINE
12	End of day
13	Dynamic storage configuration
14	INPUT or RDBACK data set activity
15	OUTPUT, UPDAT,INOUT,OUTIN data set activity
17	Scratch data set status
18	Rename data set status
19	Direct access volume
20	Job commencement
21	ESV
30	Start TS (information sur TSO-time-sharing sous OS)
31	TIOC
32	Driver

33	Driver modify
34	Step termination (pour TSO)
35	Logoff
38	Initial TS configuration
40	Dynamic DD
41	Modify TS
42	Stop TS

Nature des informations recueillies

- Informations système
  - . opérations du système
  - . E/S des unités
- Informations sur les fichiers (data-set)
- Informations sur les volumes
  - . utilisation de l'espace sur les volumes à accès direct
- Informations sur les "job" et les "step"

Nous nous sommes intéressé aux enregistrements de code 4. Cet enregistrement apporte tout renseignement sur un "step" du JCL d'OS [8]

- Code de l'enregistrement
- heure de fin du "step"
- date de fin du "step"
- identification du programmeur
- identification du système
- nom du "Job"
- heure de départ du "Job"
- n°du "step"
- heure de départ du "step"
- date de départ du "step"
- nombre de cartes SYSIN de l'utilisateur

- Code de retour du "step"
- Nom du programme exécuté (nom du processeur)
- Nom du "step"

  ⋮ (Configuration de la mémoire durant le "step")

- Temps CPU

  ⋮

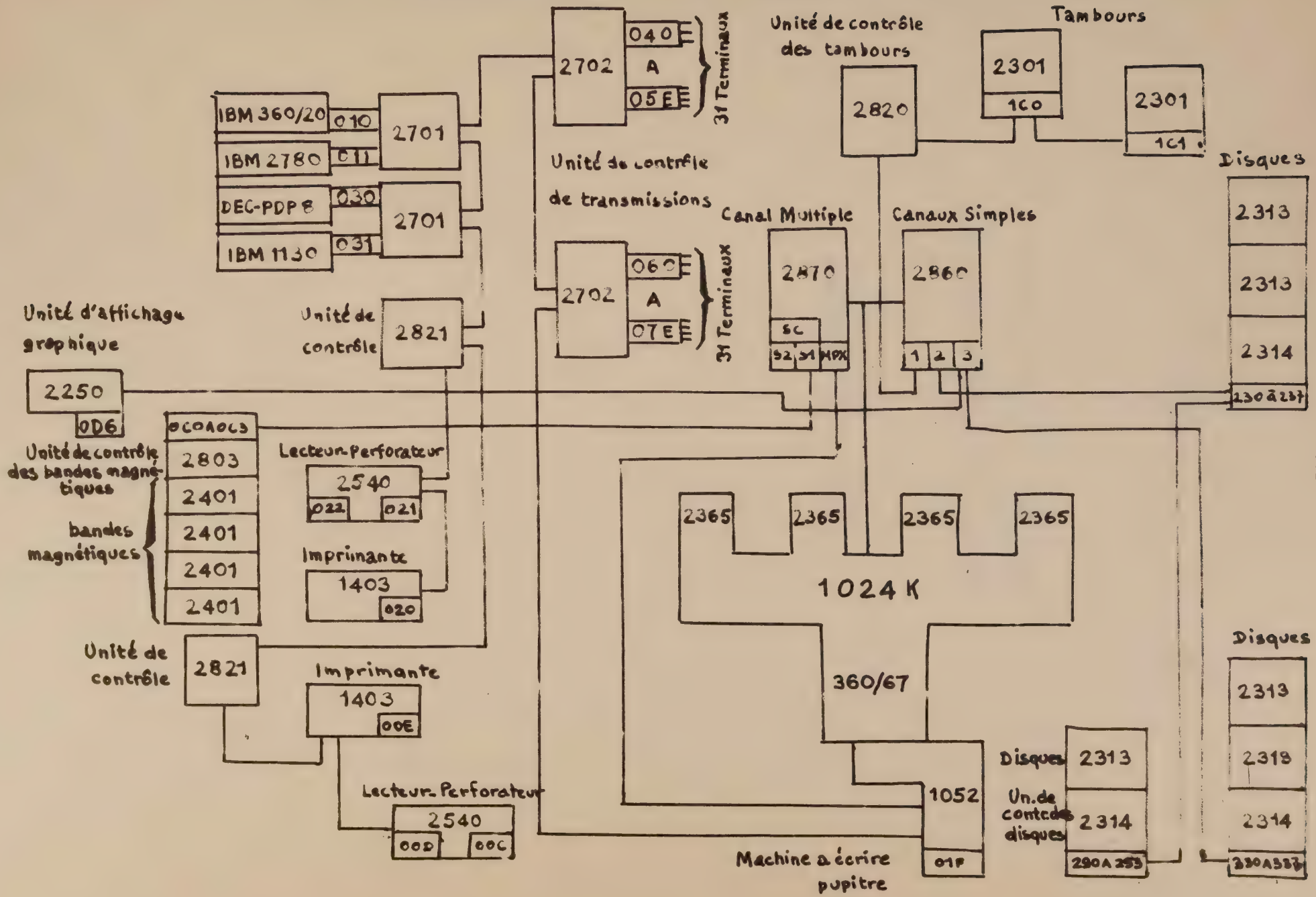
  (Configuration des périphériques et des fichiers utilisés durant le "step")

A N N E X E 3

<>*****<>

CONFIGURATION DE L'I.M.A.G.GRENOBLE

CONFIGURATION I.M.A.G.



A N N E X E 4

<>*****<>

RESULTATS POUR LES 2 ECHANTILLONS

DE MESURES PRELEVEES DANS CP/CMS

- 11-03-72 au 6-04-72
- 30-06-72 au 19-07-72

RESULTATS DES CALCULS STATISTIQUES EFFECTUES SUR LES MESURES PRISES PAR CMSMESUR

RESULTATS GLOBAUX (EN JOURS)

Résultats pour la période du 11.03.72 au 6.04.72

POURCENTAGE DE COMMANDES FAUSSES = 2.212

POURCENTAGE DE COMMANDES ABREGEES = 23.813

TEMPS MOYEN SEPARANT 2 COMMANDES DANS LE SYSTEME = 3.37SEC. ECART-TYPE = 6.76

MOYENNE DES I/O CONSOLE PAR COMMANDE = 9.95 ECART-TYPE = 37.14

MOYENNE DES I/O DISQUE PAR COMMANDE = 81.90 ECART-TYPE = 312.72

MOYENNE DU NOMBRE D'UTILISATEURS DU SYSTEME = 34.464 ECART-TYPE = 7.520

TAILLE DE L ECHANTILLON TRAITE = 59734 COMMANDES

NOMBRE DE MACHINES DANS LE SYSTEME = 256

NOMBRE DE COMMANDES TRAITEES = 67

TABLEAU DES COMMANDES

Fréquence de la commande ↓ — Longueur moyenne fichier → — Fréquence d'accès ↓ — nb E/S disque — nb E/S console — ↓ E/S disque — ↓ E/S console — % d'une même commande — Pourcentage de commandes provenant

NOM	% COMDE.	MOY-LONG	NB.FICH	% FREQUR	MOY. IOD	MOY. IOC	E-C LONG	E-C IOD	E-C IOC	% EXEC
ALTER	3.874 %	0.0	0.0	11.57 %	6.46	0.10	0.0	4.91	0.30	73.66 %
CLOSIO	1.358 %	0.0	0.0	0.0 %	0.45	0.02	0.0	2.34	0.20	45.37 %
COMBINE	1.246 %	507.51	387.00	6.32 %	108.47	0.44	790.40	157.65	0.68	53.96 %
ERASE	9.085 %	504.69	4362.00	21.18 %	7.63	0.03	1169.11	29.39	0.27	44.05 %
FILEDEF	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
FINIS	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
LISTF	3.052 %	0.0	0.0	15.47 %	6.43	0.13	0.0	17.12	22.07	1.43 %
OFFLINE	8.330 %	226.70	4441.00	6.37 %	30.72	0.20	474.74	54.34	0.79	38.87 %
PRINTF	1.656 %	107.41	889.00	13.43 %	6.00	11.50	606.37	0.30	30.73	14.31 %
SPLIT	0.581 %	0.0	0.0	0.0 %	82.72	1.26	0.0	111.61	0.66	62.54 %
STATE	5.301 %	172.11	2059.00	34.97 %	0.0	0.01	386.23	0.0	0.08	99.97 %
TAPE	2.379 %	0.0	0.0	6.76 %	52.01	2.30	0.0	193.20	11.60	19.35 %
EDIT	11.927 %	254.95	6387.00	2.46 %	61.35	30.47	421.08	106.54	51.52	4.18 %
CEDIT	0.045 %	0.0	0.0	0.0 %	513.44	36.63	0.0	534.78	51.89	0.0 %
UPDATE	0.293 %	83.91	93.00	9.92 %	194.83	0.76	85.57	113.17	0.61	90.91 %
GENMOD	0.591 %	0.0	0.0	3.12 %	114.67	0.14	0.0	101.57	0.35	75.64 %
GLOBAL	6.537 %	0.0	0.0	2.37 %	1.82	0.03	0.0	2.35	0.20	99.88 %
LOAD	4.963 %	127.43	2863.00	15.62 %	157.50	0.93	179.77	165.42	10.31	43.32 %
LOADMOD	0.233 %	5525.53	138.00	0.72 %	11.53	0.07	2572.69	10.63	0.29	40.20 %
USE	0.561 %	0.0	0.0	77.31 %	102.50	1.43	0.0	75.05	1.60	91.34 %
REUSE	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
START	2.811 %	0.0	0.0	22.28 %	95.71	43.47	0.0	206.37	90.53	35.38 %
IS	0.670 %	103.30	371.00	25.75 %	184.40	33.56	103.66	205.66	80.93	9.25 %
CLRCVER	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
DEBUG	0.569 %	0.0	0.0	0.0 %	0.46	4.09	0.0	3.28	3.36	22.35 %
ASSEMBLE	0.164 %	163.26	89.00	37.76 %	664.14	3.43	240.68	478.78	4.29	7.14 %
ASMG	1.122 %	523.84	663.00	24.73 %	1242.17	8.78	619.35	1266.61	15.96	24.18 %

ALGOL	0.040 %	268.92	24.00	33.33 %	252.50	4.63	93.47	69.54	4.78	54.17 %
PLI	1.117 %	264.04	657.00	53.29 %	723.34	11.19	455.63	323.73	13.16	19.34 %
FORTRAN	1.622 %	158.94	942.00	13.37 %	55.46	2.72	192.14	58.67	4.96	11.04 %
PL360NEW	0.049 %	88.37	27.00	41.38 %	33.62	3.38	83.32	24.45	5.82	0.0 %
SNOBOL	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
SCRIPT	0.285 %	461.58	161.00	21.18 %	27.34	60.33	383.15	23.11	154.78	2.35 %
AWX	0.407 %	0.0	0.0	36.75 %	136.07	3.63	0.0	33.30	6.35	0.0 %
FORTLIB	0.261 %	0.0	0.0	37.82 %	66.40	3.47	0.0	33.12	4.07	0.0 %
COBOL	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
MODULE	2.574 %	4336.15	1537.00	24.72 %	223.44	39.65	2581.71	643.06	91.09	46.79 %
BR	0.005 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
CPFUNCTN	4.516 %	0.0	0.0	11.53 %	1.80	0.07	0.0	3.26	0.57	[illegible]
LOGIN	3.103 %	0.0	0.0	3.91 %	6.83	0.62	0.0	5.53	0.49	26.17 %
RET	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
	1.184 %	0.0	0.0	0.0 %	14.61	1.35	0.0	9.33	4.65	0.0 %
BLIF	4.533 %	0.0	0.0	0.0 %	0.04	0.0	0.0	0.03	0.0	96.46 %
STAT	1.711 %	0.0	0.0	3.58 %	4.03	0.97	0.0	4.67	0.30	3.51 %
DISK	0.132 %	0.0	0.0	4.12 %	43.99	0.68	0.0	69.66	2.04	31.58 %
RELEASE	0.214 %	0.0	0.0	17.97 %	2.66	0.25	0.0	3.95	0.65	0.0 %
PLANNING	0.330 %	0.0	0.0	17.77 %	4.93	0.22	0.0	5.66	0.45	0.0 %
LINEND	0.647 %	0.0	0.0	0.0 %	0.10	0.0	0.0	0.69	0.0	94.47 %
MODMAP	0.002 %	0.0	0.0	0.0 %	33.00	1.00	0.0	0.0	0.0	0.0 %
FORMAT	0.018 %	0.0	0.0	9.09 %	7.82	5.64	0.0	0.66	1.12	0.0 %
READTAP	0.072 %	0.0	0.0	4.65 %	250.25	22.70	0.0	600.68	31.80	0.0 %
DUMPF	0.129 %	0.0	0.0	2.50 %	5.85	8.73	0.0	3.93	14.66	0.0 %
SORT	0.059 %	0.0	0.0	2.86 %	381.23	7.37	0.0	380.49	12.91	46.[illegible] %
CHARDEF	0.164 %	0.0	0.0	1.10 %	0.39	0.09	0.0	2.83	0.65	91.23 %
TAPDEF	0.030 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
TXTLIB	0.229 %	0.0	0.0	13.14 %	93.30	0.84	0.0	146.45	1.89	8.70 %
MACLIB	0.060 %	0.0	0.0	11.11 %	54.25	3.50	0.0	49.45	12.54	0.0 %
WTAPCS	0.030 %	0.0	0.0	5.56 %	143.72	34.00	0.0	168.72	57.52	0.0 %

IWRTAPE	0.033 %	0.0	0.0	5.00 %	33.73	0.25	0.0	27.61	0.55	75.00 %
ILTSPCMS	0.007 %	0.0	0.0	0.0 %	23.25	56.75	0.0	3.40	64.08	0.0 %
ISPACE	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
ITPCOPY	0.007 %	0.0	0.0	25.00 %	4.75	5.00	0.0	1.50	6.27	0.0 %
IECHO	0.005 %	0.0	0.0	0.0 %	2.00	26.33	0.0	0.0	32.15	0.0 %
IBRUIN	0.030 %	0.0	0.0	0.0 %	72.33	159.44	0.0	97.83	146.49	0.0 %
IPILB	0.020 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
IDUMPFT	0.013 %	0.0	0.0	0.0 %	10.83	0.38	0.0	7.45	0.52	0.0 %
ISYN	0.012 %	0.0	0.0	0.0 %	3.43	3.57	0.0	1.51	7.76	0.0 %
ITAPEIO	0.007 %	0.0	0.0	0.0 %	0.75	0.0	0.0	1.50	0.0	75.00 %
IGLOAD	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
IARBRF	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
ICENWAIT	0.007 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	100.00 %
ISIMUL	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
ICCMPARE	0.002 %	0.0	0.0	0.0 %	14.00	1.00	0.0	0.0	0.0	0.0 %

TABLEAU DES CATEGORIES

	CAT.1	CAT.2	CAT.3	CAT.4	CAT.5
% DES COMMANDES	136.903 %	112.174 %	116.335 %	6.367 %	121.691 %
% DES ERREURS	115.445 %	2.572 %	112.774 %	140.135 %	6.963 %
MOYENNE DES TCC	18.65	65.23	77.61	491.77	34.06
MOYENNE DES TCC	1.45	30.00	6.85	9.32	5.47
ECART-TYPE TCC	71.37	116.53	143.07	736.57	237.39
ECART-TYPE TCC	10.11	51.23	44.59	49.16	34.96

Pourcentage d'erreurs global

5,70
0,31
2,16
2,03
1,54

11,74 % d'erreurs

Catégorie 1 manipulations de fichiers.

Catégorie 2 éditions de fichiers.

Catégorie 3 exécutions - changements - mises au point

Catégorie 4 assemblages et compilations

Catégorie 5 divers.

TABLEAU DES FICHIERS

TYPE	% FICH.	MOY LONG	E-C LONG
SYSIN	14.614 %	564.50	647.29
UPDATE	4.221 %	95.29	405.43
ALGOL	0.586 %	209.30	146.38
ALGOLW	2.264 %	295.91	336.19
PLI	12.304 %	263.13	419.35
FORTRAN	16.020 %	179.21	269.31
COBOL	0.125 %	47.05	23.38
SNOBOL	0.0 %	0.0	0.0
SCRIPT	2.287 %	420.59	474.93
EXEC	4.295 %	18.44	78.34
TEXT	32.772 %	102.62	156.72
MODULE	10.501 %	4446.57	2610.30

TABLE DES MACROCOMMANDES

TYPE DE MACROCDE.	% CDES.	% ERREUR	MOY-LONG	NB. FICH	MOY. IOD	MOY. IOC	E-C LONG	E-C IOD	E-C IOC
<nom macro> TYPE IMPLICITE	4.060 %	0.825 %	18.52	2425.00	413.02	30.07	120.79	765.99	70.52
EXEC <nom macro> TYPE EXPLICITE	3.170 %	0.158 %	11.89	1891.00	153.42	6.74	7.99	501.56	35.92

FREQUENCES DES NIVEAUX DE RECURSION

I NIV.1	I NIV.2	I NIV.3	I NIV.4	I NIV.5	I NIV.6	I NIV.7	I NIV.8	I NIV.9 I
I 0.439811	I 0.105651	I 0.065591	I 0.033671	I 0.008711	I 0.000871	I 0.0	I 0.0	I 0.0 I

Fréquences cumulées

44% des commandes

proviennent de macro-commandes.

RESULTATS DES CALCULS STATISTIQUES EFFECTUES SUR LES MESURES PRISES PAR [illegible]

RESULTATS GLOBAUX (4 JOURS)

Résultats pour la période du 30.06.72 au 19.07.72

POURCENTAGE DE COMMANDES EMISES = 1.[illegible]

POURCENTAGE DE COMMANDES ABREGEES = 23.[illegible]
TEMPS MOYEN SEPARANT 2 COMMANDES DANS LE SYSTEME = 3.57 SEC. ECART-TYPE = 5.24

MOYENNE DES E/S CONSOLE PAR COMMANDE = [illegible] ECART-TYPE = 37.84

MOYENNE DES E/S DISQUE PAR COMMANDE = [illegible] ECART-TYPE = 372.41

MOYENNE DU NOMBRE D'UTILISATEURS DU SYSTEME = 25.357 ECART-TYPE = 6.222
TAILLE DE L'ECHANTILLON TRAITE = [illegible] COMMANDES
NOMBRE DE VACATIONS DANS LE SYSTEME = 221
NOMBRE DE COMMANDES TRAITEES = [illegible]

TABLEAU DES COMMANDES

Fréquence de la commande ↓ — Longueur moyenne fichier — Fréquence d'erreur ↓ — $\overline{nb}$ E/S disque — $\overline{nb}$ E/S console — σ E/S disque — σ E/S console — Pourcentage de commandes provenant ↓ d'une macro-commande

NOM	% COMM.	MOY-LONG	NB.FICH	% ERREUR	MOY. IOD	MOY. IOC	E-C LONG	E-C IOD	E-C IOC	% EXEC
ALTER	2,333 %	0.0	0.0	0.07 %	6.36	0.00	0.0	5.03	0.29	66.92 %
CLOSIO	1,532 %	0.0	0.0	0.0 %	0.27	0.01	0.0	2.10	0.19	59.78 %
COMBINE	2,246 %	261,33	379.00	7.17 %	94.11	0.43	[illegible]	171.31	0.75	31.66 %
ERASE	9,151 %	510.16	2513.00	20.81 %	8.03	0.03	1073.17	17.07	0.17	47.35 %
FILEDEF	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
FINIS	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
LISTF	3,914 %	0.0	0.0	20.06 %	5.53	0.37	0.0	9.27	13.76	5.30 %
OFFLINE	0,325 %	200,19	2373.00	5.42 %	37.23	0.20	622.49	75.56	1.03	30.13 %
PRINTF	1,534 %	64.32	314.00	14.07 %	6.23	11.44	262.71	7.63	23.31	17.35 %
SPLIT	0,553 %	0.0	0.0	0.0 %	69.39	1.10	0.0	112.22	0.72	83.96 %
STATE	1.315 %	699.11	309.00	51.42 %	0.0	0.0	1426.34	0.0	0.0	30.18 %
TAPE	4.059 %	0.0	0.0	10.76 %	80.13	2.96	0.0	402.99	17.56	19.30 %
EDIT	11,656 %	202.06	2341.00	3.74 %	74.74	32.62	357.90	176.46	53.58	6.19 %
CEDIT	0,102 %	0.0	0.0	5.71 %	308.33	19.23	0.0	375.74	24.24	25.71 %
UPDATE	0,229 %	0.0	0.0	17.70 %	76.52	0.27	0.0	42.19	0.44	99.12 %
GENMOD	0,341 %	0.0	0.0	3.53 %	107.73	0.13	0.0	133.66	0.34	64.10 %
GLOBAL	5,936 %	0.0	0.0	0.79 %	1.77	0.02	0.0	1.71	0.16	90.45 %
LOAD	4,552 %	132,53	1454.00	10.23 %	147.79	1.76	187.24	144.19	16.14	49.15 %
LCARDMOD	0,498 %	7595.05	170.00	1.17 %	2.62	0.03	2526.22	5.51	0.17	23.04 %
USE	0,239 %	0.0	0.0	72.41 %	93.11	1.53	0.0	73.27	1.62	81.90 %
RELSE	0,003 %	0.0	0.0	0.0 %	15.00	0.0	0.0	0.0	0.0	0.0 %
START	3,677 %	0.0	0.0	20.56 %	161.52	42.16	0.0	501.07	89.18	47.40 %
$	0,499 %	0.0	0.0	22.31 %	183.02	43.05	0.0	139.47	78.39	5.85 %
CPROVER	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
DEBUG	0,629 %	0.0	0.0	0.0 %	0.0	3.28	0.0	0.0	2.29	25.00 %
ASSEMBLE	0,129 %	1034.22	41.00	15.31 %	1459.11	5.80	1142.64	1250.71	9.81	0.09 %
ASMG	3,135 %	668.75	387.00	21.02 %	1250.33	10.11	336.92	1414.91	37.61	39.49 %

IALGOL	0.312 %	124.47	3.00	23.0 %	103.25	1.25	176.09	118.26	0.56	75.00 %
IPLI	0.637 %	227.19	215.60	54.81 %	763.37	34.01	559.56	289.34	16.62	36.37 %
IFORTRAN	1.595 %	160.17	325.00	13.13 %	55.40	2.51	165.59	31.17	4.67	10.04 %
IPLI360NEW	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
ISNOBOL	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
ISCRIPT	0.128 %	294.62	34.00	34.34 %	22.77	27.23	443.34	22.95	91.26	0.0 %
IAUX	0.232 %	255.14	79.00	34.23 %	133.40	4.29	333.10	105.95	4.66	0.0 %
IFORT18	0.370 %	0.0	0.0	10.14 %	59.49	3.85	0.0	25.82	7.28	0.0 %
ICOBOL	0.0 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	0.0 %
IMODULE	0.113 %	2961.37	724.00	13.11 %	249.27	57.10	2433.60	721.29	97.55	38.29 %
IPLIP	4.059 %	0.0	0.0	0.0 %	0.31	0.09	0.0	0.24	0.02	97.72 %
I	1.313 %	0.0	0.0	0.0 %	21.46	1.95	0.0	0.23	7.64	0.0 %
ILOGIN	3.522 %	0.0	0.0	0.30 %	5.47	0.60	0.0	3.72	0.49	47.27 %
ICPFUNCTN	4.359 %	0.0	0.0	3.73 %	1.37	0.12	0.0	3.27	0.81	47.57 %
ISTAT	0.617 %	0.0	0.0	0.33 %	2.32	0.76	0.0	4.51	0.49	28.81 %
ICHARDEF	1.196 %	0.0	0.0	0.0 %	0.32	0.01	0.0	0.67	0.15	97.11 %
IREACTAP	0.119 %	0.0	0.0	14.52 %	77.33	16.33	0.0	150.19	17.36	0.0 %
ILINEND	0.652 %	0.0	0.0	0.0 %	0.11	0.09	0.0	1.10	0.07	85.71 %
IDUMPEI	0.594 %	0.0	0.0	21.59 %	12.43	1.34	0.0	14.95	0.86	41.38 %
ISORT	0.099 %	0.0	0.0	3.43 %	194.71	2.93	0.0	446.55	1.67	67.74 %
IDUMPE	0.070 %	0.0	0.0	4.22 %	10.48	12.03	0.0	12.76	12.69	0.0 %
IDISK	0.939 %	0.0	0.0	5.56 %	91.37	0.77	0.0	176.74	3.33	22.47 %
IKT	0.012 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
IDESRUE	0.015 %	0.0	0.0	0.0 %	0.0	0.0	0.0	0.0	0.0	30.00 %
II	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
IDELCASE	0.652 %	0.0	0.0	1.34 %	1.03	0.02	0.0	0.33	0.13	91.52 %
IMACLIB	0.059 %	0.0	0.0	15.00 %	64.15	0.60	0.0	63.30	1.10	25.00 %
IWRTAPE	0.096 %	0.0	0.0	0.0 %	62.12	0.27	0.0	4.66	1.45	100.00 %
IRQUIN	0.009 %	0.0	0.0	0.0 %	23.23	117.33	0.0	16.17	123.91	33.33 %
ITAPEIO	0.179 %	0.0	0.0	6.56 %	1.11	0.13	0.0	1.84	0.46	52.46 %
IASTIGTRC	0.002 %	0.0	0.0	100.00 %	0.0	0.0	0.0	0.0	0.0	100.00 %
IPL360	0.220 %	0.0	0.0	41.37 %	74.32	1.95	0.0	195.83	2.93	0.0 %

TABLEAU DES CATEGORIES

	CAT.1	CAT.2	CAT.3	CAT.4	CAT.5
% DES COMMANDES	137.951 %	111.487 %	116.629 %	4.238 %	24.679 %
% DES ERREURS	114.343 %	3.173 %	111.421 %	32.280 %	5.697 %
MOYENNE DES IOD	29.57	76.87	95.46	526.43	29.11
MOYENNE DES IOC	1.47	31.57	10.13	7.33	4.58
ECART-TYPE IOD	147.87	179.11	244.34	941.41	227.71
ECART-TYPE IOC	3.14	57.73	46.34	29.62	31.22

Pourcentage d'erreurs globa

5,43
0,35
1,88
1,36
1,4

10,42 % d'erreurs

catégorie 1 manipulations de fichiers

catégorie 2 éditions de fichiers

catégorie 3 exécutions - changements - mises au point

catégorie 4 assemblages et compilations

catégorie 5 divers.

TABLEAU DES FICHIERS

TYPE	% FICH.	MOY LONG	E-C LONG
SYSIN	15.458 %	514.51	724.53
UPDATE	0.457 %	1334.40	1758.03
ALGOL	0.217 %	120.70	132.04
ALGOLW	2.512 %	641.40	553.04
PLI	18.063 %	213.17	134.73
FORTRAN	26.542 %	177.35	[illegible]
COBOL	0.006 %	21.34	14.47
SNOBOL	0.0 %	0.0	0.0
SCRIPT	1.270 %	411.53	473.57
EXEC	7.124 %	57.09	114.72
TEXT	20.162 %	140.99	186.53
MODULE	11.143 %	4563.30	2973.71

TABLE DES MACROCOMMANDES

	TYPE DE MACROCDE.	% CDES.	% SYSTEME	MOY-LONG	NB. FICH	MOY. IOP	MOY. IOC	E-C LONG	E-C IOP	E-C IOC
<nom macro>	TYPE IMPLICITE	4.679 %	2.41 %	[illegible]	1627.00	408.49	29.79	145.16	955.79	74.04
EXEC <nom macro>	TYPE EXPLICITE	6.437 %	2.233 %	11.53	217.00	723.19	37.63	16.76	530.33	75.44

FREQUENCES DES NIVEAUX DE REPUSSION

NIV.1	NIV.2	NIV.3	NIV.4	NIV.5	NIV.6	NIV.7	NIV.8	NIV.9
0.44198	0.07463	0.23154	0.0	0.0	0.0	0.0	0.0	0.0

↑

44% des commandes
proviennent de macro-commandes.

A N N E X E 5

<>*****<>

RESULTATS POUR L'ECHANTILLON DE MESURES

PRELEVEES DANS OS/360 MVT

- 9-03-72 au 8-04-72

JOURNEE DU 72.069 AU 72.129 9-03-72 au 8-04-72

NOMBRE DE STEPS EXECUTES	41907	
TEMPS C.P.U. TOTAL	61284695	$\times 10^{-2}$ sec
TEMPS C.P.U. MOYEN	1483	$\times 10^{-2}$ sec
NOMBRE DE LECTURES	72184	
NB DE TRANSMIT	0	
NB DE RECORD	0	

Période analogue à celle du traitement pour CP/CMS

		NOMBRE D'APPELS	% NB TOTAL D'APPELS	SOMME DES TEMPS CPU	TEMPS MOYEN	ECART TYPE	% TEMPS TOTAL	NB IMAGES DE CARTES	NB MOYEN DE CARTES	ECART TYPE	NB_OS
				10^{-2} sec	10^{-2} sec						
PREPROCS	1	0									
ALGOL	1	505	1.21	146373	289	179	0.24				
ALG141	1	0									
ALGOLX	2	595	1.66	1297671	1665	4235	2.12	0	0.00	0.00	
595											
XALGOLW	2	1	0.00	267	267	0	0.00	0	0.00	0.00	
1											
AA	2	106	0.25	52473	495	555	0.09	0	0.00	0.00	
106											
ALGOLW	2	47	0.11	5948	206	232	0.02	0	0.00	0.00	
47											
IEHASM	3	2270	5.42	1516372	644	684	3.13	5383	2.37	13.62	
2270											
ASMCASM	3	949	2.26	535557	990	760	1.53	347	0.37	1.83	
949											
MASM	3	0									
IFKAOO	4	0									
IEYFORT	4	3065	7.31	1991943	649	984	3.25	1840	0.60	5.78	
3065											
WATFOR1	4	192	0.45	135817	723	1058	0.23	0	0.00	0.00	
192											
IEQCBL00	5	1546	3.93	1281114	773	1037	2.09				
IKFCBL00	5	3	0.01	513	171	29	0.00	0	0.00	0.00	
3											
KAFAMAC	6	367	1.21	61729	151	215	0.13	0	0.00	0.00	
507											
CAGO1	7	25	0.06	85933	3436	2792	0.14	0	0.00	0.00	
25											
IFMAA	9	4766	11.37	5160486	1082	1093	8.42	0	0.00	0.00	
4766											
COMPILER	9	38	0.09	6368	167	43	0.01	0	0.00	0.00	
38											
MLISP	10	0									
LISP	10	0									
NOYAU	11	1	0.00	69	69	0	0.00	0	0.00	0.00	
1											
PL360	11	170	0.41	27134	159	238	0.04	0	0.00	0.00	
170											
SNOBOL	12	0									
IEWL	13	9420	22.48	2909321	308	11477	4.75				
LOADER	13	1216	2.90	3012978	2477	3096	4.92	0	0.00	0.00	
1216											
IKJEFF40	13	0									
DEJCSMP2	13	7	0.02	41932	5990	180	0.07	0	0.00	0.00	
7											
IMM	13	6	0.01	847	141	230	0.00	0	0.00	0.00	
6											
DELINK	13	6	0.01	260	43	14	0.00	0	0.00	0.00	
6											
DISKFACT	13	29	0.07	53455	1843	217	0.09	0	0.00	0.00	
29											
SORT	13	203	0.48	54360	267	436	0.09	0	0.00	0.00	
203											
IERRCOO0	13	368	0.73	168065	515	1450	0.25	0	0.00	0.00	
368											
REPMULT	13	67	0.16	21697	323	211	0.04	0	0.00	0.00	
67											
PGM=*.DD	13	1885	21.20	21092201	3499	13715	59.73	75	0.01	0.16	

exécutions de programmes utilisateurs

8895											
ISH	13	1408	3.35	224570	160	490	0.37	100	2.07	0.52	
1408											
133	13	2150	5.13	329720	153	603	0.54				
AUTRES	13	3216	7.67	1024900	3156	9883	16.72	6662	2.07	11.74	
3216											

GROUPE	NOMBRE D'APPELS	% NB TOTAL D'APPELS	SOMME DES TEMPS CPU	TEMPS MOYEN	ECART TYPE	% TEMPS TOTAL	NB IMAGES DE CARTES	NB MOYEN DE CARTES	ECART TYPE	NB_OS
ALGOL60	505	1.21	146373	289	179	0.24				
ALGOL W	849	2.03	1359659	1601	3879	2.22	0	0.00	0.00	
849 ASSEMBL	3219	7.68	2856820	887	710	4.66	5730	1.78	11.52	
3219 FORTRAN	3257	7.77	2131760	654	939	3.48	1840	0.57	5.61	
3257 COBOL	1649	3.93	1281627	777	1056	2.09				
FORMAC	507	1.21	81729	161	215	0.13	0	0.00	0.00	
507 GPSS	25	0.06	85760	3430	2792	0.14	0	0.00	0.00	
25 PL/I	4766	11.37	5160436	1082	1093	8.42	0	0.00	0.00	
4766 PPS	38	0.09	6363	167	43	0.01	0	0.00	0.00	
38 LISP	0									
PL360	171	0.41	27203	159	237	0.04	0	0.00	0.00	
171 SNOBOL4	0									
UTILITY	26521	64.24	40132567	1738	13368	73.57	8806	0.33	3.98	2
6921										

A N N E X E 6

<>*****<>

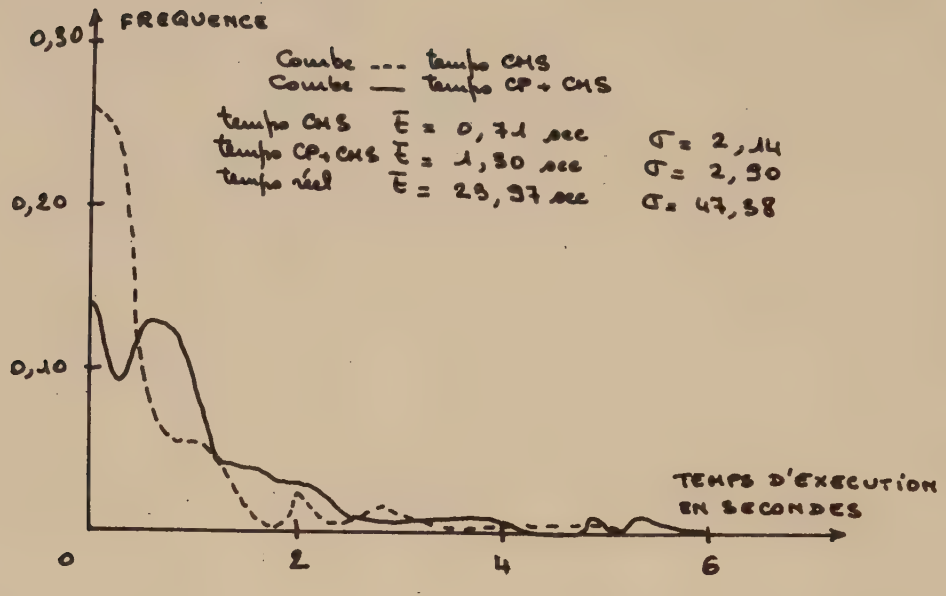
RESULTATS COMPLEMENTAIRES SUR L'ANALYSE

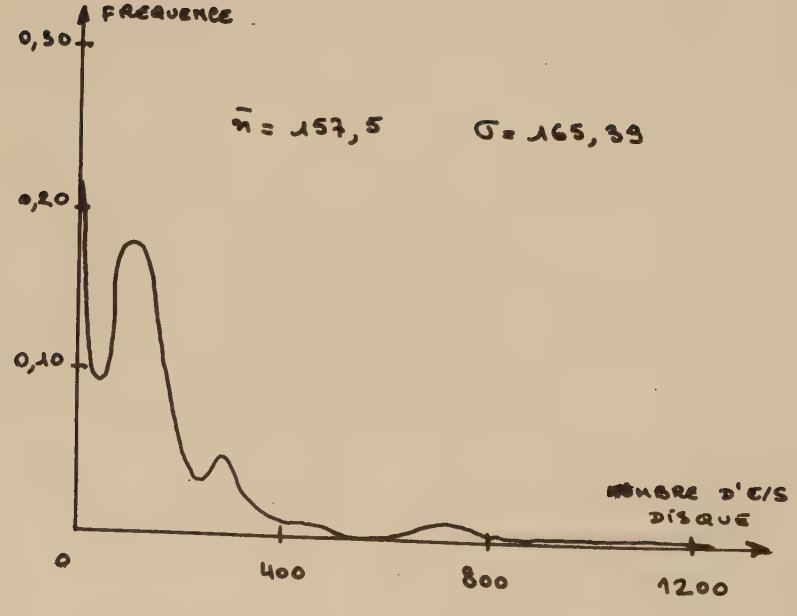
DES MESURES DANS CP/CMS

Commande	Fréquence %	% temps virtuel CMS	% temps U.C. (CP)	% temps réel	Overhead %	temps virtuel moyen	Nb moyen E/S disque	Nb moyen E/S consoles	
EDIT	11,92	4,96	9,14	51,35	65,13	0,49	61,35	30,47	
ERASE	9,08	0,14	0,24	0,15	61,81	0,01	7,68	0,03	
OFFLINE	8,33	2,90	4,88	1,42	61,81	0,41	30,72	0,20	
GLOBAL	6,53	0,08	0,10	0,07	49,11	0,01	1,82	0,03	
STATE	5,30	0,07	0,06	0,00	13,52	0,01	0,00	0,01	
LOAD	4,96	3,00	3,52	1,88	45,35	0,71	157,50	0,93	
BLIP	4,93	-	-	-	-	-	-	-	
CPFUNCTN	4,51	-	-	-	-	-	-	-	
ALTER	3,87	0,05	0,09	0,05	62,60	0,01	6,46	0,10	
LOGIN	3,10	-	-	-	-	-	-	-	
LISTF	3,05	0,12	0,36	1,35	77,45	0,04	6,49	8,13	
START	2,81	27,41	21,13	11,69	16,67	11,48	95,71	48,45	
Module	2,57	14,87	16,18	12,17	40,96	6,81	223,44	39,65	
TAPE	2,37	0,40	1,07	0,85	75,74	0,20	52,04	2,30	
STAT	1,71	-	-	-	-	-	-	-	
PRINTF	1,69	0,08	0,21	0,93	75,40	0,05	6,00	11,50	
FORTRAN	1,62	7,16	5,51	1,41	16,50	5,20	56,46	2,72	
CLOSIO	1,35	0,01	0,02	0,01	54,83	0,01	0,45	0,02	
COMBINE	1,24	0,74	0,77	0,28	38,34	0,69	108,47	0,44	
Login implicite	1,18	-	-	-	-	-	-	-	
ASMG	1,12	10,43	11,86	4,10	43,50	10,95	1242,17	8,78	
PLI	1,11	12,03	11,06	4,15	30,12	12,69	723,84	11,19	
$	0,67	9,03	6,73	2,47	13,81	15,88	184,40	33,56	

COMMANDE "LOAD" : CHARGEMENT DE PROGRAMMES ET EDITION DE LIENS

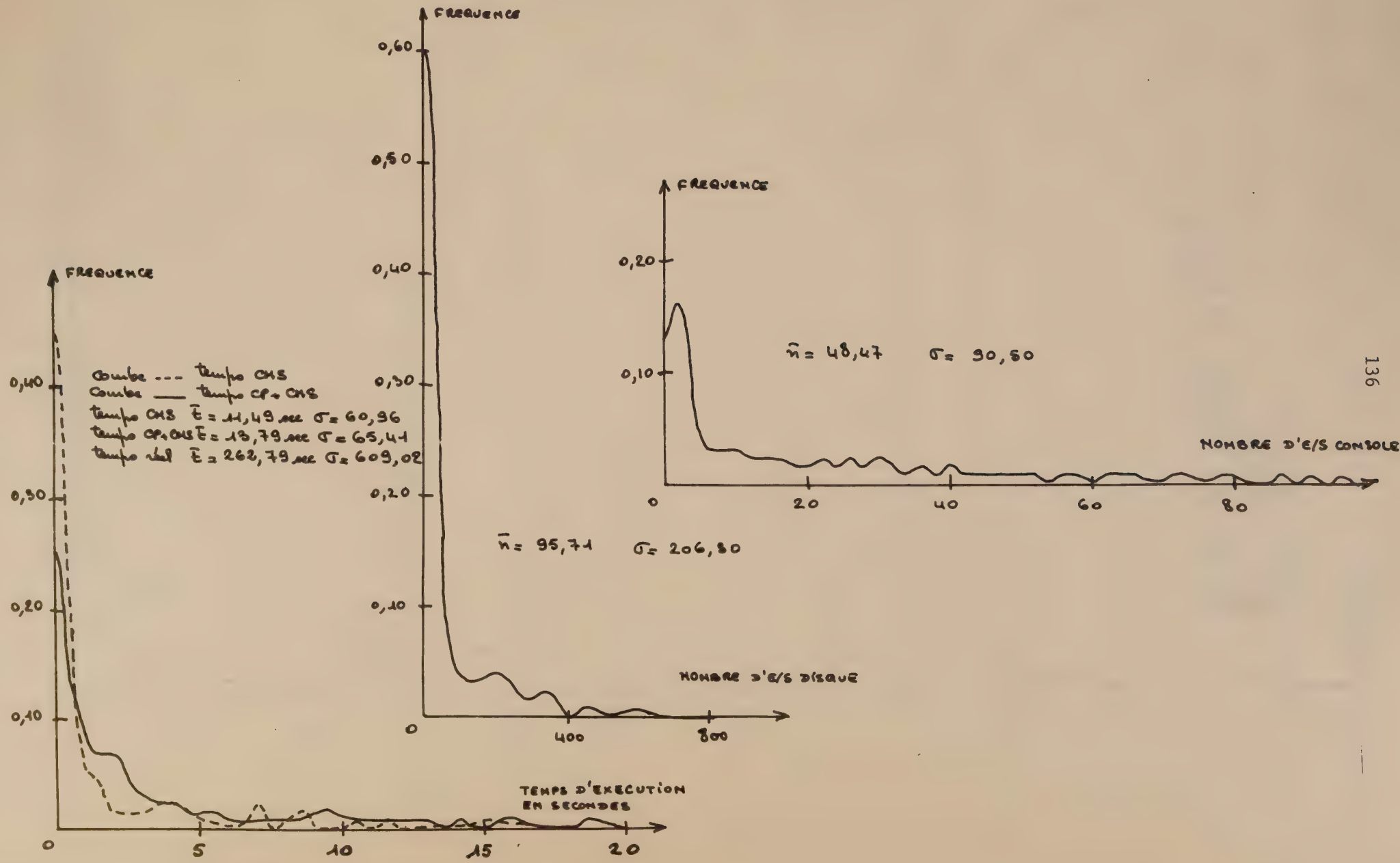
TAILLE DE L'ECHANTILLON 2964



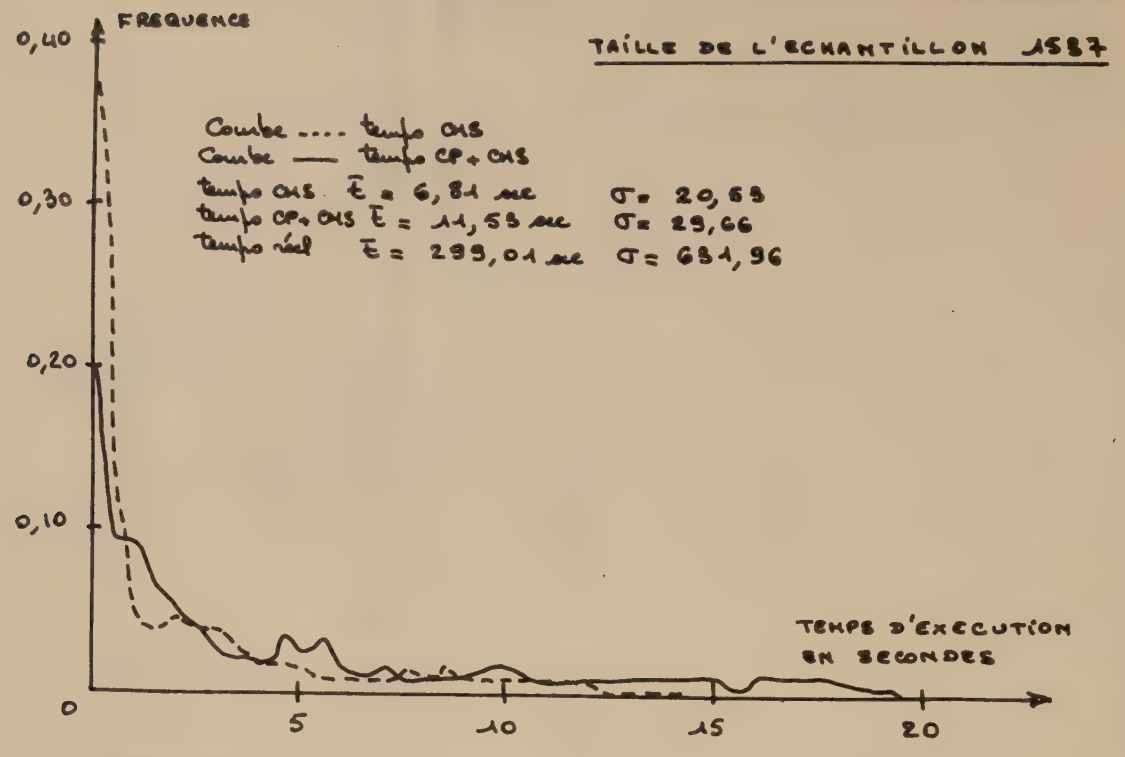


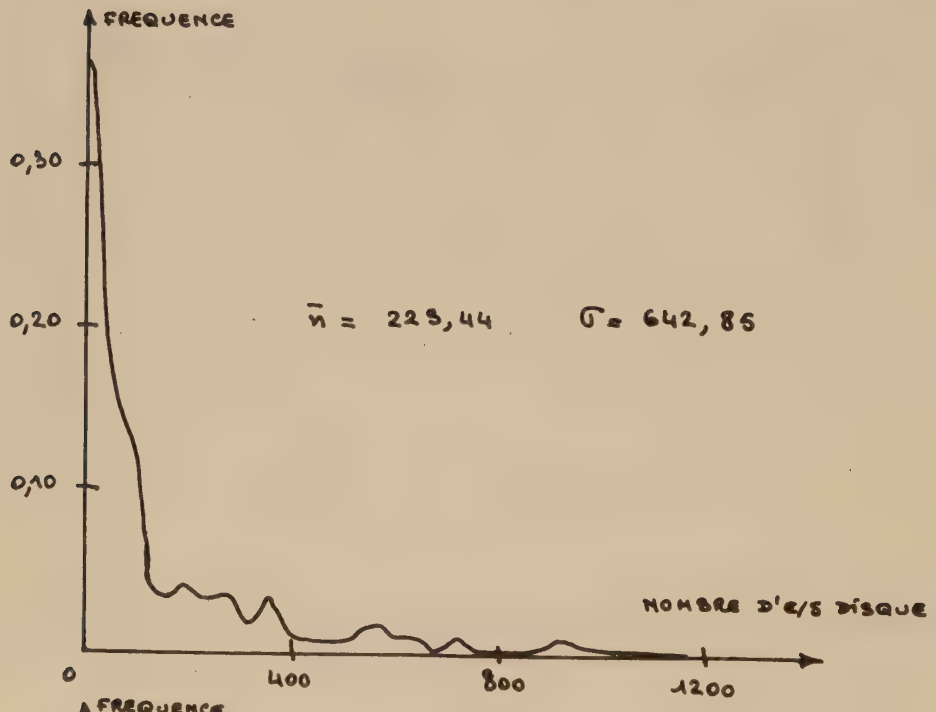
COMMANDE "START" : EXECUTION DE PROGRAMMES UTILISATEUR

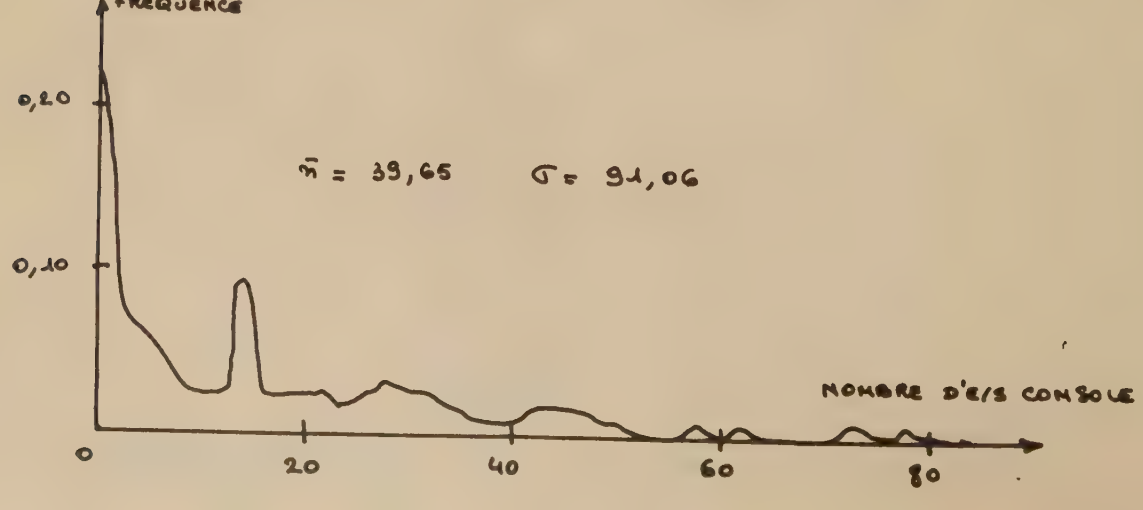
TAILLE DE L'ECHANTILLON 1679

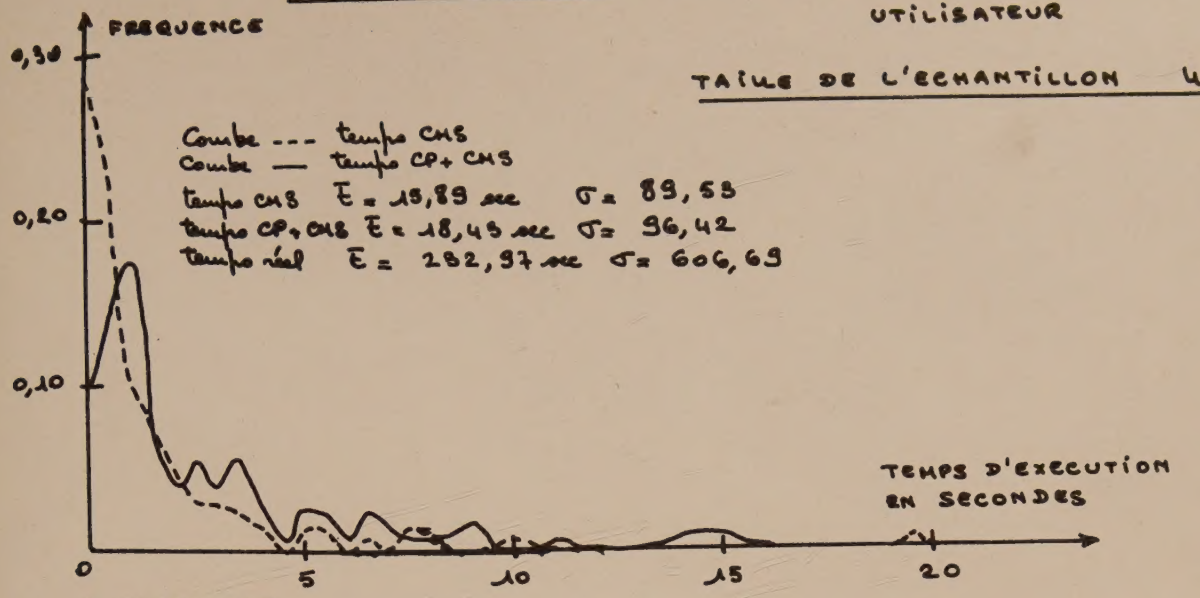
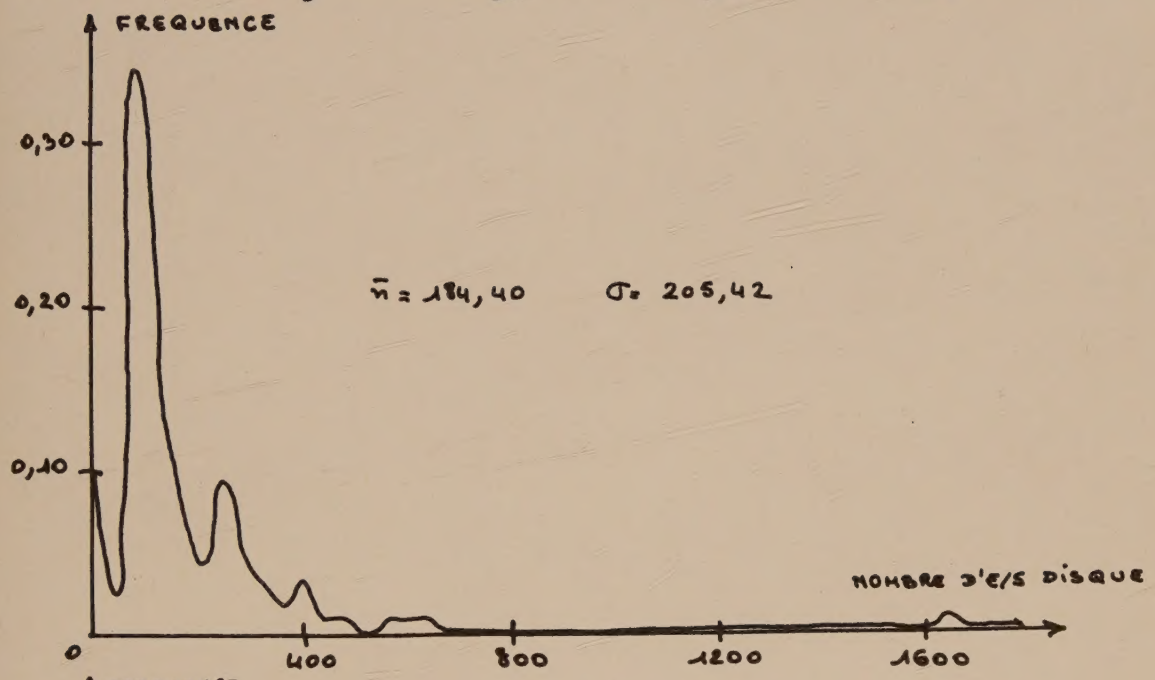
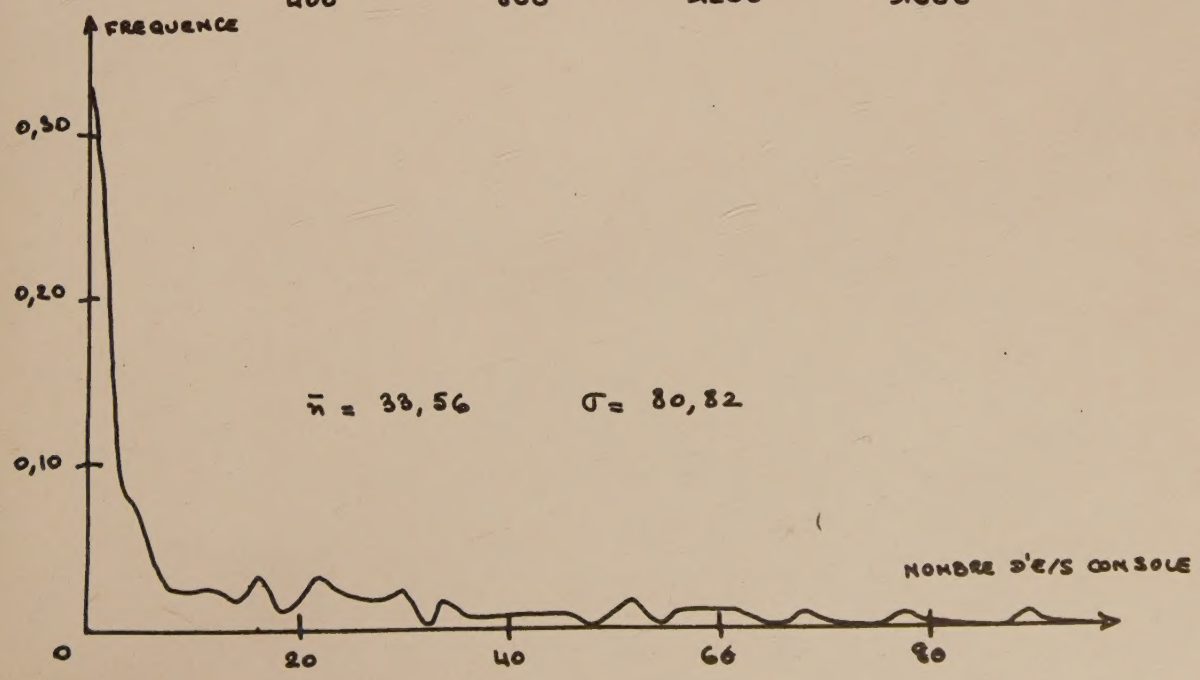


COMMANDE D'EXECUTION DE MODULES D'UTILISATEUR







COMMANDE "$" : CHARGEMENT ET EXECUTION DE PROGRAMME UTILISATEUR

TAILLE DE L'ECHANTILLON 400